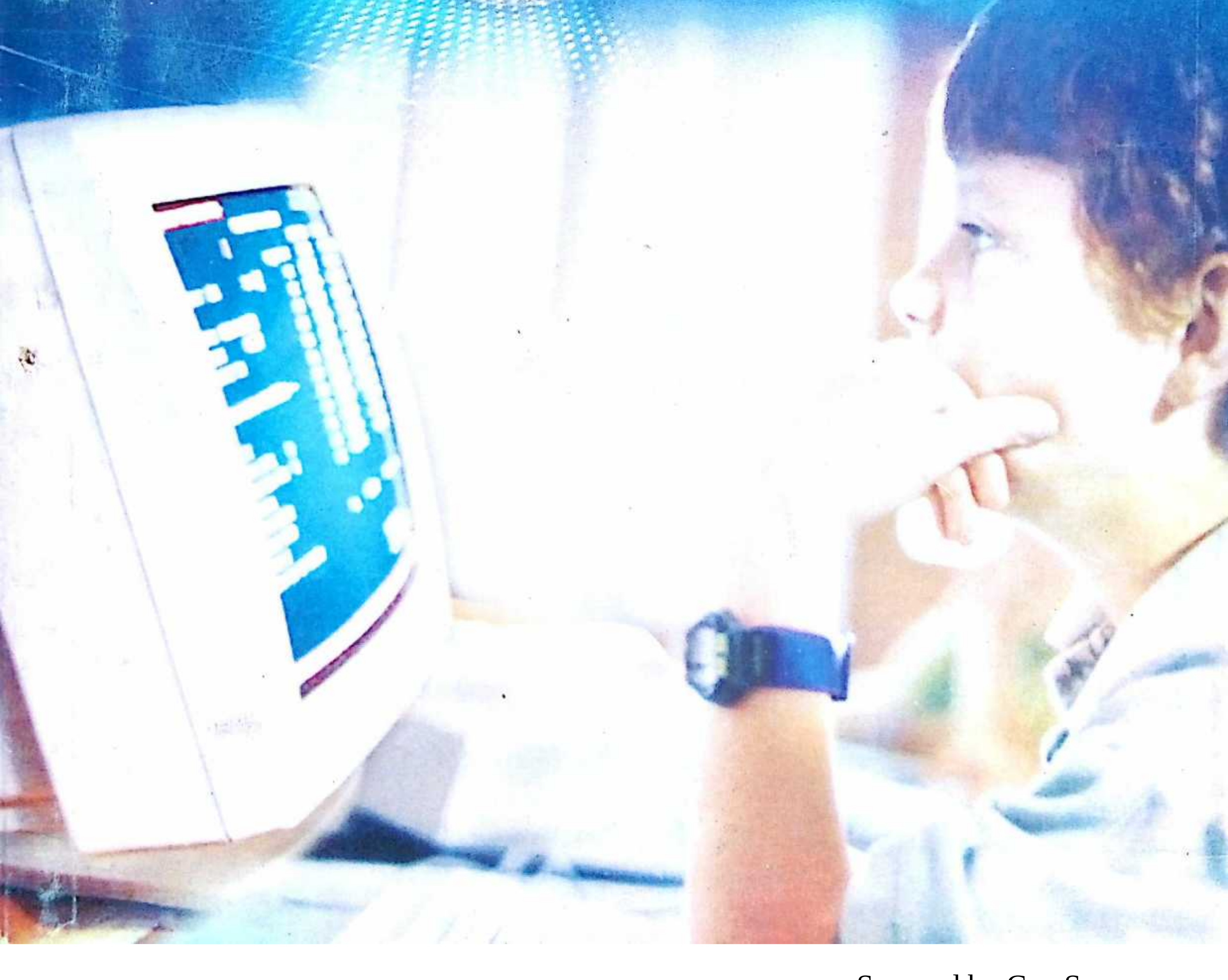
A text book of

COMPUTER SCIENCE

For Class IX & X

By
Prof. Muhammad Tahir Hassan



A TEXTBOOK OF

COMPUTER SCIENCE

Revised Edition 2003

for

CLASS IX & X

by

Prof. Muhammad Tahir Hassan

Digitized By M. Y. M. B

National Book Foundation

Islamabad

Lahore-Rawalpindi-Multan-Bahawalpur-Faisalabad-Karachi-Sukkur
Hyderabad-Larkana-Peshawar-Abbottabad-Quetta-D.I.Khan-Bannu

CONTENTS

PART-I
INTRODUCTION TO COMPUTER SCIENCE

PART-II
INTRODUCTION TO BASIC PROGRAMMING LANGUAGE

PREFACE TO THE FIRST EDITION

The past few years have witnessed an explosive growth in the use of computers, because of the indispensable role that computers play in our daily life. The microprocessor technology has opened a new world of automation and has radically changed the life-style. In a developing country like Pakistan, where the literacy rate is very low, every educated person needs an exposure to the serious use of computer, ignorance of which would cause a national crisis. Therefore, it is paramount to take drastic step in the application and education of computers to ensure the bright future of our country.

The introduction of computer science as an optional subject at secondary and higher secondary school level is a first step in this direction. This book is designed to fulfill the need of curriculum developed by the National Curriculum Development Committee.

The text has been divided into two parts. Part I is an Introduction to computer science consisting of six chapters and Part II is on BASIC Programming language consisting of seven chapters. In preparing the book, care has been taken to use simple language understandable by a secondary school student so as to convey the message properly and easily. Efforts have been made to minimize the errors and omissions. This will surely enhance the confidence and understanding of the students in the subject. Yet there is always room for improvement and any suggestion and criticism to improve the quality of this work will be warmly welcomed.

It is by the grace of Almighty Allah, the prayers of my parents and my wife that enabled me to complete the book. The author is highly indebted to Dr. Fakhar-ul-Islam Lodhi for his valuable suggestions to improve the text and for reviewing the script in spite of his heavy commitments. The author is extremely thankful to members of the National review committee for their encouraging remarks and valuable suggestions to improve the script. The author also wishes to acknowledge the efforts of all those who assisted him to at various stages in preparing the textbook.

Prof. Muhammad Tahir Hassan
February, 1997.

PREFACE TO THE REVISED EDITION 2001

The Computer revolution is constantly redefining and widening its course with amazing speed and has deep impact upon us as individuals and as members of society. Curriculum development is a continuous process and it is particularly true for Computer Science. First edition of Computer Studies was based on earlier curriculum in computer science that has the course contents of preliminary nature. The latest curriculum of Computer Science for higher school students, approved by the Federal Ministry of Education, Islamabad is an effort to prepare students to enter a world full of thrill, challenges and exhilaration by introducing the latest advancements both in hardware and software. It is important to make the students aware of the impact of computer in their lives since the world today has become heavily dependent on computers.

The 21st century is the harbinger of information technology (IT) through which people are getting closer as the world is squeezing into a global village. The developing countries have also realized the need of focusing more on information technology to bring them at par with the first world. The spread of IT is impossible without taking measures to introduce computers at grass roots level.

Pakistan is also shaping out a work plan to establish IT institutions from city to town level. In this regard, work on an IT city is under way. If the government's claim about IT revolution in the country meets a success, the country would very soon appear prominently on the horizon of IT-led nations with its venerable economy and a prosperous future generation.

To be successful each and every high school student must be computer literate. Once the students understand the importance of being able to use computers, the second objective is to help them make the transition from being computer literate to being computer user with the emphasis to realize it as a mind tool. In the process they will learn what a computer can and cannot do and how to use computers to solve problems and accomplish tasks. The computer learning curve is steep and the progress can be slow but once they are on the proper track, they would be able to reach the dazzling heights of this curve where a promising future is awaiting them.

The material covered in the second edition has a broad scope of coverage and there is a balance among technology, application, and programming. The textbook of Computer Studies has been changed in numerous ways from the last edition to reflect the changes that have occurred in the syllabus of Computer

Science. The book has thoroughly been revised and enlarged to meet the objectives of the latest curriculum. In the light of the new syllabus, some of the chapters have been restructured; some have been modified while additional topics have been presented in the following chapters:-

Chapter 6 on BOOLEAN ALGEBRA —— that provides the foundation in understanding the logic used in digital computers.

Chapter 8 on WINDOWS —— a very powerful yet easy-to-use microcomputer operating system without which applications like Word processing, Excel worksheet, Access, power point Paint program etc. cannot run on your computer. This chapter provides understanding that will help the students to get the most out of their computers.

Chapter 19 on WORD PROCESSING —— the most popular Word processing software lets you easily create, edit, format, and store documents like your assignments, letters, etc.

And Chapter 18 on GRAPHICS —— includes draw and fill software in word processing software or paint program. This provides you a way to explore software packages on graphics to draw charts / graphs or piece of art design.

Chapters 6 and 8 are included in Part I for class IX while chapters 18 and 19 are included in Part II for class X. It is hoped that the introduction of these chapters in the second edition would make it also acceptable by the beginners in the field of Computer. I have tried to make this an error free edition. As there is always room for improvement, suggestions and criticism leading to improve the quality of the material presented in the book will be highly appreciated.

It is worth mentioning here the support that National Book Foundation, Islamabad provided me in developing this book. I am highly indebted to Mr. Hafeez Tauqir, Director Production, NBF, Islamabad, for his encouragement and help at every step during the development of this edition. At the end I would like to acknowledge the efforts of all those who assisted me in preparing the second edition particularly my son Mr Naveed Tahir in typing the script and in drawing the figures and to my daughter Miss Hina Tahir in reading the proof and making valuable suggestions. It is by the grace of Almighty Allah, who blessed me the courage and energy to complete this task.

Prof. Muhammad Tahir Hassan
April 19, 2001.

PREFACE TO THE REVISED EDITION 2003

The second edition on Computer Studies had been developed with the objective that it will not only fulfill the need of studying students but also serve the purpose of teacher's manual as well. In this respect it was a composite book on the subject which had been found useful for students who want to study the topics in detail.

Present edition is a revised second edition which has been developed to incorporate the suggestions of the reviewers. The name of present edition is changed as **A Text Book on Computer Science** instead of A Text Book on Computer Studies. The objective to develop this edition was to make the contents as simple as possible and making it brief required at SSC level according to the demand of the syllabus. Some of the topics not included in the course outline such as electrical relay circuits form Chapter 6 on Boolean Algebra and Chapter 18 on graphic packages. The topics related to the computer laboratory work have been excluded form the text.

> A separate **Computer Lab. Journal** has been developed according to the syllabus 2000 on Computer Sciene as suggested by the reviewers, since no proper lab. Journal was available. This Computer Lab. Journal contains labs on Computer's essentials, DOS, Windows, and Programming in BASIC using assignment statements, input/output statement, Control statements, arrays, Sub program, file handling, graphic statements and computer labs on MS–Word in a very simple and practical way. It is hoped that this journal will provide the students a deep insight of the topics and enable them to carry out computer labs. conveniently and effectively.

It has become more vital to revise the text book with the introduction of new system of evaluation to test the understanding of the subject among students. For this purpose model objective questions have been provided. These questions will also guide the teachers and the paper setters to develop their own objective questions. Moreover summary of each chapter has also been given at the end of each chapter. This will provide answers to most of the objective as well as short answered questions.

Glossary of important terms has been included at the end of text book. An important feature of this revised edition is the reduction and simplicity of the material presented in it. The name of the title has also been changed as pointed out by the reviewers chapter. This provides answers to many objective as well as short as short answered questions.

Efforts have been made to produce an error free edition. However suggestions and criticisms will be highly appreciated. I am keenly looking for the comments and criticisms on the book from the readers, so that further improvement may be made wherever necessary.

It is not out of place the support given to me by the NBF, Islamabad especially Mr. Muhammad Aslam Rao, Secretary NBF and Mr. Hafeez Tauqir, Director Production, NBF Islamabad in introducing the previous edition throughout the country. I am also extremely thankful to the National Review Committee for their suggestions, healthy criticism and encouragement that boosted me to put all of my effort to make it simple and comprehensive. At the end I would like to acknowledge the efforts of all those who assisted me in preparing the present edition.

I do admit that I could not do this job without the help and the grace of Almighty Allah. It is He Who blessed me the courage and energy to complete the task. It is also the prayers of my parents, particularly my mother who left this world praying all the time for my success.

Prof. Muhammad Tahir Hassan
February 05, 2003

1

INTRODUCTION TO COMPUTER

Man is always after finding ways to ease his physical and mental labour. He succeeded in making machines that could ease his mental labour. Early machines could simply add, subtract, multiply and divide numbers. Computers have got the present form after modifications and refinements that continued through centuries. Modern computers have grown up as the most fast and powerful tool as a result of the technological innovations during the last fifty years. Rapid progress in electronics has made it possible to bring it within the reach of everyone.

The Computer has enabled man to conquer space and is helping him in every sphere of life. The present age is the information age. Whether you are a teacher or a student, a doctor or a patient, a pilot or an engineer, a scholar or a businessman, the information super-highway fulfils the need of everyone.

1.1 HISTORY AND DEVELOPMENT OF COMPUTER

The Computer has evolved through centuries. From the time the man first started using arithmetic, he had been inventing devices to handle numbers. One of the earliest and the simplest computing device was **ABACUS** developed around

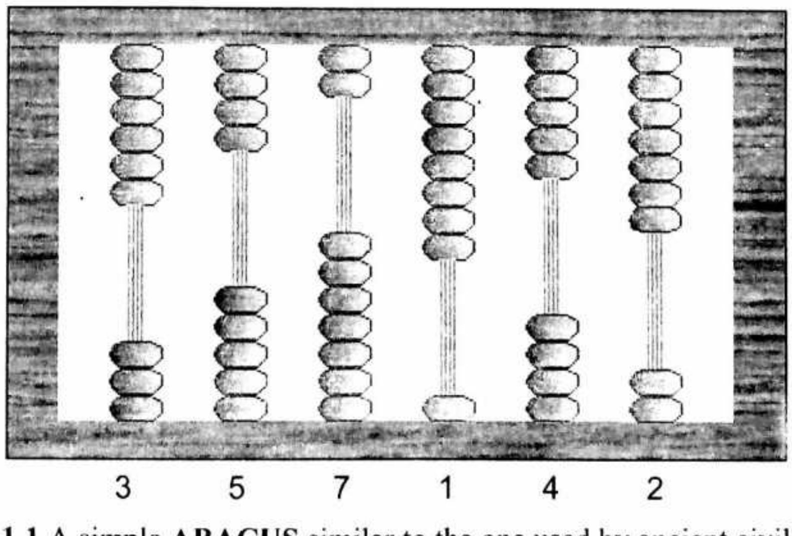


Figure 1.1 A simple **ABACUS** similar to the one used by ancient civilizations

3000 BC. A simple ABACUS is a rectangular wooden frame carrying several parallel wires. Each wire supports number of beads as shown in Figure 1.1. These beads are free to slide along the length of the wires when counting, adding etc. Each bead below represents a digit.

1.2 MECHANICAL COMPUTING DEVICES

One of the first recognized calculating devices was the Abacus. It is still being used in its many forms in countries such as Japan, China, and Russia. Users can perform calculations almost as quickly as people do with calculators.

In 1614 John Napier invented logarithms and a device called Napier's bones or rods. The rods helped to multiply numbers. These rods not only simplified tedious calculations but also were fast and accurate.

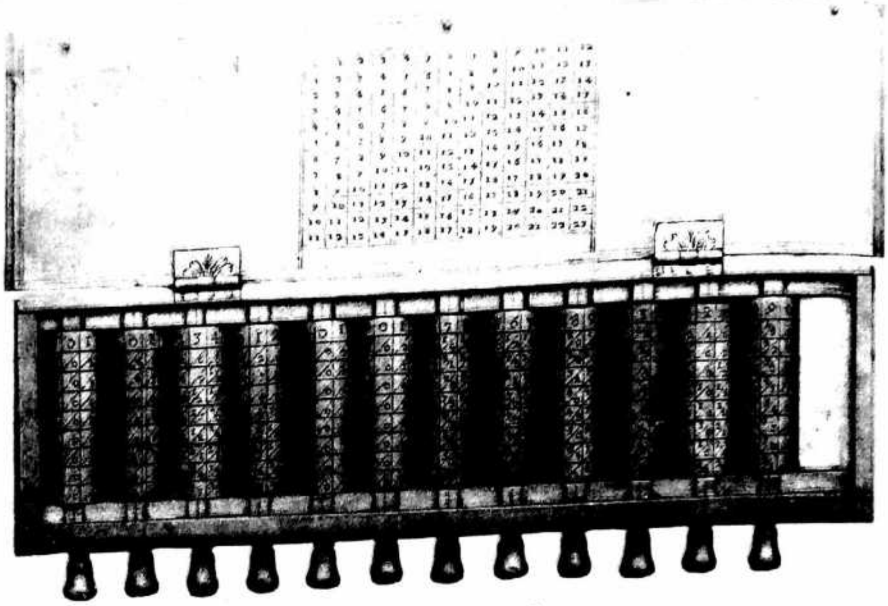


Figure 1.2 Napier's bones

Napier's idea of logarithm soon gave birth to another very useful calculating device called **Slide Rule**. A slide rule consists of two scales; the **Rule** and the **Slide** and a transparent rectangular moving piece called **cursor** as shown in Figure 1.3. The cursor is capable to slide smoothly over the slide rule.

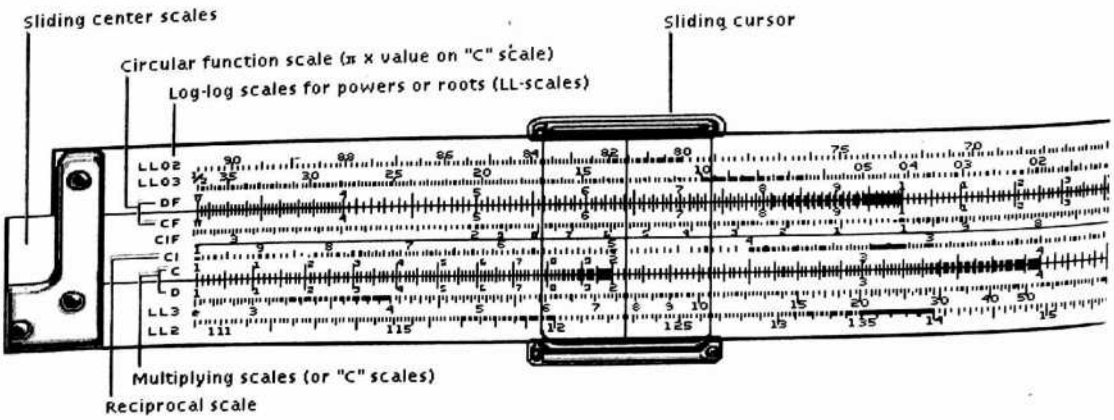


Figure 1.3 Slide Rule

The next attempt in the field of automatic computation was made in 1642 by 19 years old Blaise Pascal. He developed a machine known as Pascaline that could add and subtract. It consisted of a series of wheels or dials. Each dial had 10 digits from 0 to 9 on it. These dials were geared with drums also having digits from 0 to 9 which were displayed through windows as shown in Figure 1.4. Pascaline had the ability to carry digits to the next drum.

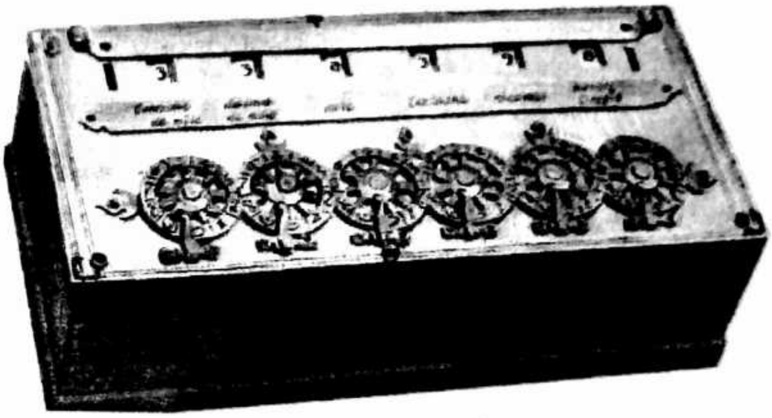


Figure 1.4 Pascaline; Pascal's adding machine

In 1672 Leibnitz developed a calculating machine that could not only add and subtract but multiply and divide also. In 1790 Joseph Maria Jacquard built a loom. It used vertical rods to pull up the threads. Punched cards were used that allowed some of the rods to pass through the holes blocking the remaining rods. Punched cards were joined in a series and were passed over the rods one after the other. The shuttle movements created patterns on fabric woven on the loom. The pattern was repeated after each cyclic rotation of the punched cards. Jacquard's invention emphasized three important concepts.

- Coding of information by punching holes on the cards.
  (Storage concept)
- Linking of the cards in a series to provide instructions in sequence.
  (Programming concept)
- Job would be performed automatically as the program run.
  (Program execution concept)

In 1786 J. H. Muller proposed a calculating machine called a **difference engine.** The idea of the difference engine was conceived by Charles Babbage in 1812. Babbage began to develop a difference engine but after 19 years of labour, he stopped working on difference engine. He began working on a new machine called the **Analytical Engine**. It could be programmed to evaluate a wide range of arithmetical expressions. The machine had to be fully automatic. Babbage's design included several features present in today's computer. His **Analytical Engine** consisted of five units shown in Figure 1.5. It had the following parts:-

Store: *This part had to store the numbers entered into the machine and those, which had to be generated at the time of processing.*

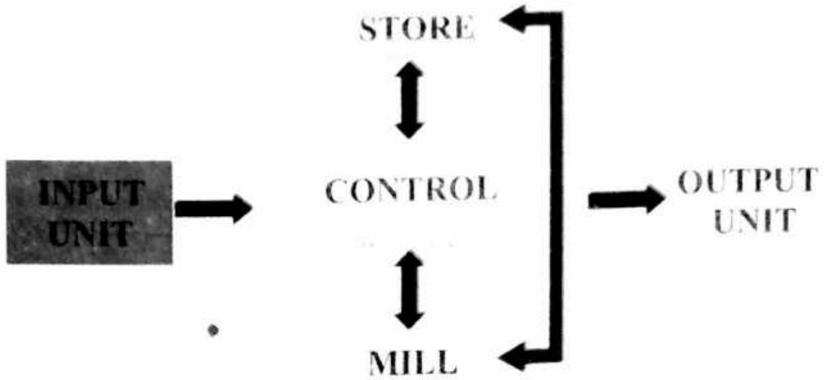


Figure 1.5 Schematic diagram of Analytical Engine

Mill: *This was the Processing unit that had to perform all the Arithmetic operations automatically by rotation of gears and wheels.*

Control: *Control unit had to control all the other units. Moreover it had to transfer the numbers and instructions from the store to the mill and vice versa, through the rotations of gears and wheels.*

Input: *Input unit had to enter data and instructions into the store. The input media had to be in the form of punched cards.*

Output: *The output unit had to display the results of calculations.*

Babbage's analytical engine could not be completed due to the under-developed technology of his time. But he laid the foundation stone for the development of modern **digital computers**. He is called the **father** of modern digital computer.

In 1880's Dr. Herman Hollerith was working at the statistics department of US Census Bureau. He was inspired by the Jacquard's idea of punched cards. In 1887 Dr. Herman Hollerith designed a Tabulating machine for 1890 census. His Tabulating Machine could read and sorted out data from punched cards.

1.3 ELECTROMECHANICAL COMPUTING MACHINES

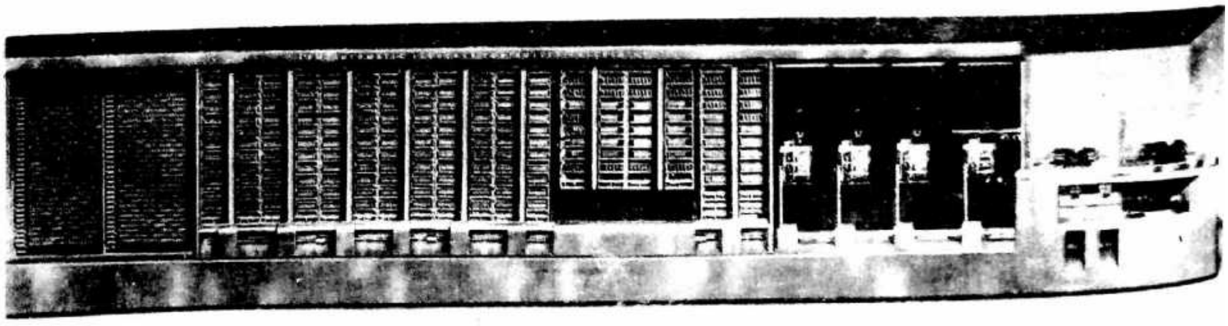


Figure 1.6 Automatic sequence controlled calculator (MARK-I)

Mechanical calculators were heavy, large and slow. The first computer that used electromagnetic relays was the complex **calculator.** It was the first machine that used **binary number system**. Another early electromechanical machine designed by Dr. Howard Aiken in 1944 was **Automatic Sequence Controlled Calculator** (ASCC) or **Mark-I** shown in figure 1.6. Mark-I was 15.55 metres long and weighed 5 tons. It used 3300 electromagnetic relays. This machine remained in operation till 1959.

1.4 GENERATIONS OF COMPUTER

The invention of Vacuum tube in 1906 opened the gates for the development of **electronic computers**. Vacuum tubes were used as electronic switches instead of relays in earlier electronic computers. Vacuum tubes were noiseless and could work at a fantastic speed as compared to relays.

The development of electronic computers can be divided into generations, depending upon the technologies used. The generations of computers are:-

- **First generation** (Vacuum tube1906) 1942 - 1959
- **Second generation** (Transistors1947) 1959 -1965
- **Third generation** (Integrated circuits IC's 1965) 1965 - 1973
- **Fourth generation** (Microprocessors 1971) Since 1972
- **Fifth generation** (Artificial Intelligence) In progress since1980s

FIRST GENERATION COMPUTERS

First generation computers used vacuum tube technology. In 1943, Electronic Numerical Integrator And Calculator **(ENIAC)** was built at University of Pennsylvania. It contained nearly 18,000 tubes, and consumed 150 kilowatt of power. ENIAC was very big and weighed more than 30 tons. It could perform 5,000 additions and hundreds of multiplications per second.

Vacuum Tube

The concept of stored program was presented by **Dr. John Von Neumann**. The idea of stored program was much the same as suggested by Babbage for his Analytical Engine. Neumann's Stored program concept pointed out the way the computer systems could function in future. The first computer that used

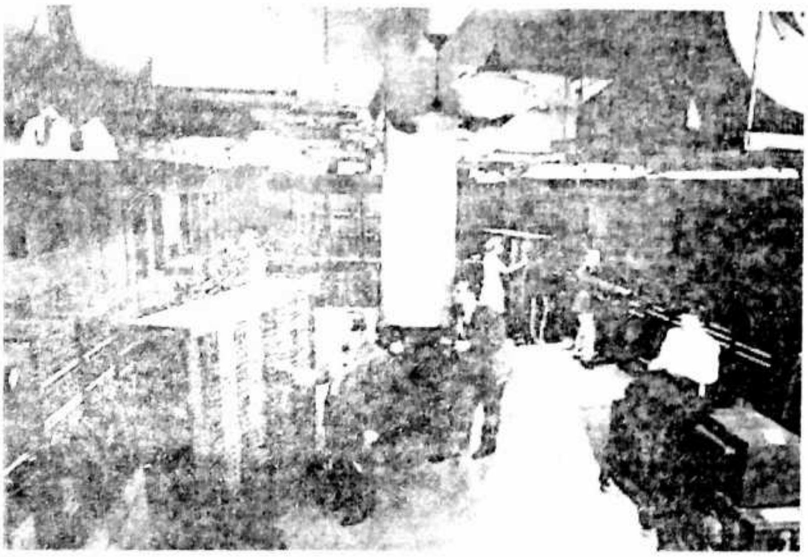


Figure 1.7 Electronic Numerical Integrated and Calculator **(ENIAC)**

stored program concept was the **Electronic Delayed Storage Automatic Computer (EDSAC)** developed in 1949. Another was the **Electronic Discrete Variable Automatic Computer (EDVAC)** that used the same concept. The program and data were fed in the EDVAC through punched paper tapes.

The first commercially successful machine was **Universal Automatic Computer** (UNIVAC-**I**) shown in figure 1.8. It was made operational in 1951. It was a self checking computer and could work 24 hours a day. UNIVAC-I used magnetic tape as input media.

Figure 1.8 UNIVAC, The first commercially successful First generation computer

The first generation computers had the following shortcomings:-

i) *Very big in size*
ii) *Slow in speed*
iii) *Low reliability*
iv) *Large power consumption and*
v) *Difficult maintenance*

SECOND GENERATION COMPUTERS

A Transistor

Transistor was invented in 1948. As compared to vacuum tubes transistors are smaller, more reliable and consume far less Power. Second generation computers were developed using transistors that were able to perform a single operations in microseconds and were capable to store huge data.

Computer manufacturers began to develop computers with more efficient storage and faster input and output capabilities. A second generation computer is shown in Figure 1.9. These computers

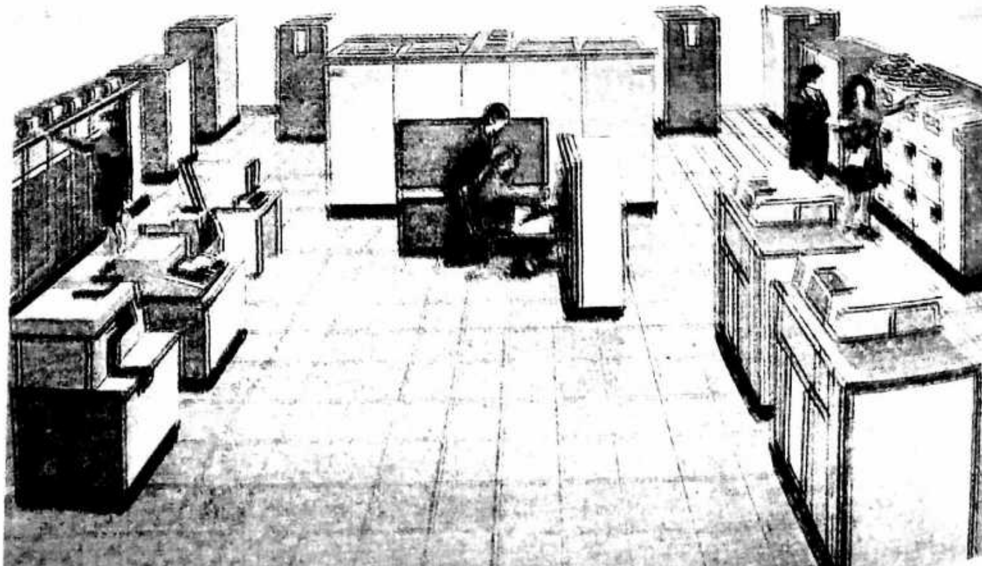
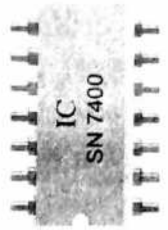


Figure 1.9 IBM 704 System

were extremely reliable, compact in size, and less heat problems. A number of hardware improvements accompanied the second-generation computers such as card readers, magnetic tapes, magnetic disks, printers etc. These machines used computer Languages similar to simple English for computer programming.

THIRD GENERATION COMPUTERS

In the sixties, the **integrated circuits (ICs.)** appeared with the development of microelectronics technology. An IC chip contains many components (such as transistors, diodes, and resistances) interconnected with each other.



Integrated circuits

Third generation computers have smaller size and consumed very little power, but their speed and storage capacity had largely been increased using integrated circuits. Figure 1.10 shows a third generation IBM computer model 85 of its System/360 series. These computers used more versatile programs like real-time programming, multiprogramming and database management. Other developments that accompanied are minicomputers and remote terminals.


Figure 1.10 An IBM System/360 Model 85 Computer System

Minicomputers had nearly the same capabilities as large computers but were smaller in size with less storage capacity. Remote terminals are computer terminals that are linked with the main computer set at some far off place.


Figure 1.11 A third generation minicomputer

FOURTH GENERATION COMPUTERS

Very Large Scale Integrated circuits (**VLSI**) were developed in 1971. A VLSI contained a complete Central Processing Unit (**CPU**) on a single Silicon Chip. The first microprocessor (μP) Intel 4004 was a 4-bit μP capable of performing arithmetical, logical and control functions. Intel developed an 8-bit Microprocessor in 1973. Later on many other companies also started producing microprocessor chips like ZX80, ZX81 etc. The use of microprocessors produced cheap computers called **Microcomputers or Personal Computers (PC)**. These computers used an 8-bit processor and the results could be displayed on TV sets.

Another popular series of personal computers was **Apple series** designed in 1976 using 8-bit microprocessors. **Commodore** and **PET** were also the popular personal computers.

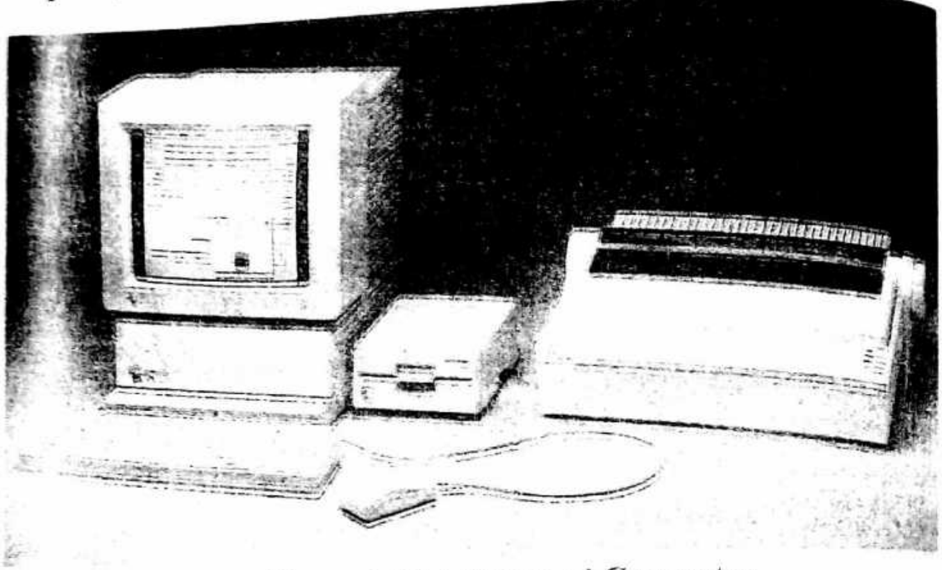


Figure 1.12 A Personal Computer

In 1978 the evolution of 16-bit microprocessors greatly increased the processing speed of the computers. A magnified sectional view of a 16-bit microprocessor is shown in figure 1.13.

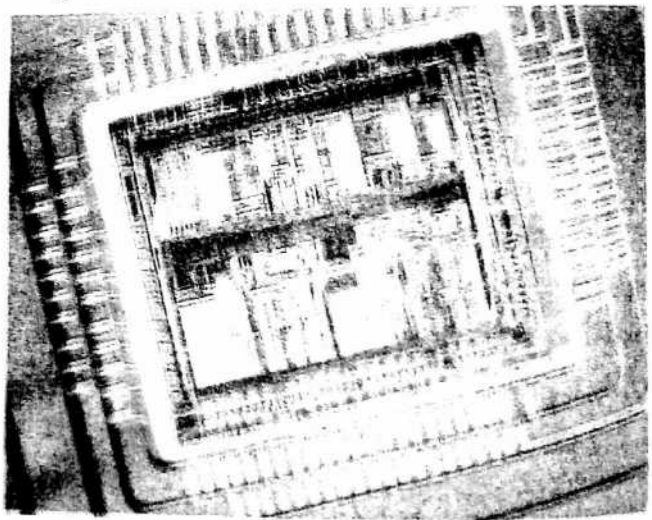


Figure 1.13 A 16-bit microprocessor

FIFTH GENERATION COMPUTERS

Till fourth generation computers, the main stress was on the improvement of hardware. This increased the speed and efficiency and reduced the size and cost of computers. However lack of **thinking power** in computers was a challenge for the scientists. The fifth generation computers are a step in this direction. These

computers will be capable of reasoning, learning, drawing inferences and making decisions. VLSI technology has currently made it possible to put millions of components onto a semiconductor chip. Improvements in computer hardware and software may produce very powerful computers than those in current use. **Artificial intelligence** and **expert systems** will be an important part of these newer fifth generation computers.

1.5 TYPES OF COMPUTER

Computers can be divided into three types. This division depends upon the design and working of a computer which differs on the type of data that enters into it and the form of its output. These three types of computers are:-

- *Analog Computers*
- *Digital Computers*
- *Hybrid Computers*

ANALOG COMPUTER

An analog computer accepts data in continuous or physical form, represents it in a suitable form to perform scientific operations. Analog data include distance, speed, pressure, temperature, fluid's flow, voltage, current etc. These are **special purpose computers** and can be used for complex scientific and mathematical calculations. These are also used in industrial units to control various processes. Analog computers are very fast in their processing but are not accurate.

DIGITAL COMPUTER

Digital computers are **general-purpose** in use. Digital computers come in many sizes and shapes. A digital computer accepts data in the form of digits represents it in terms of discrete numbers and processes numbers using various Arithmetic and logic operations. Modern digital computers are capable to store large amount of data and information and also compute data at a very high speed. They are used in almost every field of life such as scientific and technical research, business, education, health care, supermarkets, factories, banking, transportation, space exploration, art and music etc.

HYBRID COMPUTER

A hybrid computer combines best features of **analog** and **digital computers**. They are **special-purpose computers**, which are **fast** and **accurate**. Hybrid computers help the user to handle both the **analog** and **digital** data. These machines are generally used in scientific applications and in monitoring industrial processes. Hybrid computers are used in hospitals to carry out medical investigations and to watch/monitor patient's health state in intensive care unit (ICU) and alert the doctors to handle any unusual situation. These computers are also used in telemetry, spaceships, guided missiles etc.

1.6 CLASSIFICATION OF COMPUTERS

Computers can be classified according to their sizes into the following four main groups:-

- Supercomputers
- Mainframe
- Minicomputer and
- Microcomputer

This classification is based on their processing power, speed, size of main memory and other capabilities of computers.

SUPERCOMPUTERS

Supercomputers are the largest fastest and most expensive computers developed in 1980s to handle complicated problems. These have extraordinary computing power required by large organizations, scientific laboratories, aerospace centers, large industrial units, research laboratories, etc.

Supercomputers are used in space exploration forecasting weather, aircraft design, and nuclear researches. They are used in a time-sharing mode in which thousands of users can be attached to a supercomputer at the same time. Some of the supercomputers are CRAY-1, CRAY-2 and CYBER 205.

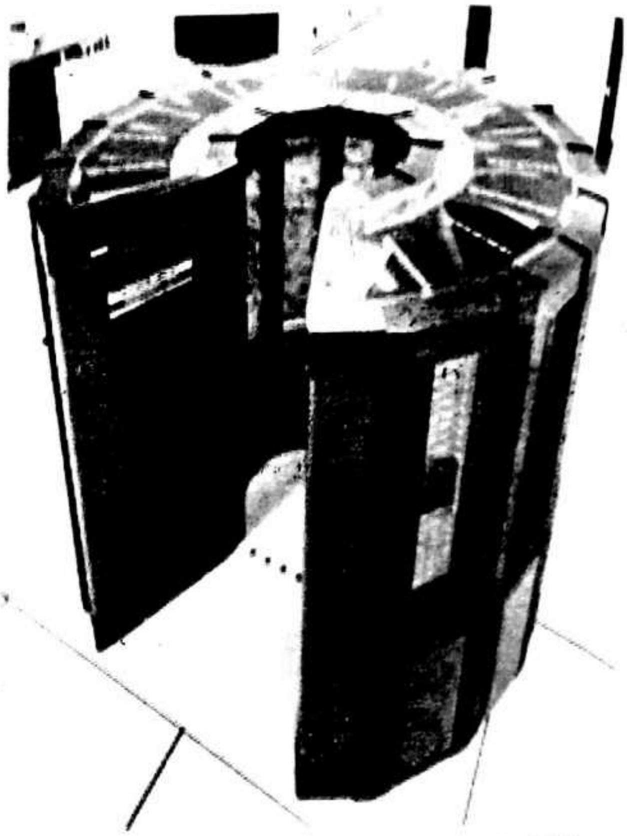


Figure 1.14 Super Computer CRAY-1

MAINFRAME

Mainframes are large-scale computers. Computer systems of this size require very large rooms and other arrangements like cooling etc. Mainframes are very fast and have very large memory. They are used by government and large organizations such as banks, airlines and universities where hundreds of users can handle multiple processing tasks at a time. This means that a mainframe is able to do different jobs for different users at the same time. For example, it is doing statistical analysis for one user and at the same time printing a report for the second user and financial statement for the third user and so on. Examples of mainframe computers are CYBER 176 and IBM 4341.

MINICOMPUTERS

These low cost computers use integrated circuits. These compact yet surprisingly powerful computers find their application in business, education and government. Minicomputers can support multiple-users and multiple tasks like mainframes through the use of terminals. Digital Equipment Corporation introduced minicomputers in the mid-1960s. Other companies such as Data General Corporation, Hewlett-Packard, IBM Corporation and Prime Computer also manufacture mini computers.

MICROCOMPUTERS

A microcomputer is a relatively inexpensive computer that is rapidly being used in all application areas. Microcomputers were introduced in 1970s as a result of the development of microprocessor. A microprocessor is a semiconductor Very Large Scale Integrated circuit (VLSI) having processing capabilities. Microprocessors are inexpensive and very small in size and thus have reduced the size and weight of microcomputers or personal computers. Microcomputers have a greater impact on some people and business than any other type of computers. Personal Computers are widely used in thousands of applications. Examples are IBM PC, AT&T, PS/2 and Apple Macintosh microcomputers.

1.7 IMPACT OF COMPUTERS AND INTERNET ON SOCIETY

As computers have acquired more features, they have acquired more uses as well. They can be used as typewriters, calculators, accounting systems, record keepers and telecommunication instruments. They can also act as canvases, tutors and toys. Microcomputers have wide variety of uses in homes. Even people with little or no programming experience can use microcomputers

Computer has emerged out as the most useful and powerful tool yet developed, which has changed the information need of most of the people all over the world. Computer technology has a deep and long-lasting influence on international trade, pattern of employment, industrial productivity, home life, entertainment, space explorations, health care, education, art, music, scientific researches and investigation of crimes etc. It has provided opportunities to many people to work and learn at home. This would bring radical changes in our society by eliminating the need of huge offices, crowded schools and universities.

COMPUTERS AND SOCIETY

With each passing day, computers are more and more involved in our lives and are affecting us in many ways. In banks, factories, stores, schools offices or even in homes we find computers or computerized systems all around us. We are tuned to use computerized microwave ovens, washing machines, VCRs etc. We also used to do computerized bank transactions, computerized shopping etc.

Computers make us more productive in many of our jobs. In education they can help us for better understanding faster learning and broaden our thinking. In hospitals we have better diagnosis, proper treatment and better healthcare. Thus computers have changed the way we work, the way we learn and the way we communicate. It is helping us to do thing efficiently and accurately.

We are also facing problems arising from the use of computers. These are unemployment due to automation, wastage of time and energy in useless computer activities, data security, personal privacy and computer crime. Technology is a tool that is neutral between good and evil. We have used it for both. We have untested technologies of tremendous power in our hands. It is too early to tell whether they will be used to build or to destroy.

THE INTERNET

Internet is a system of worldwide computer networks that enables the internet users to exchange information. Computers can be hooked to telephone through **modems** so that data can be sent and received over telephone lines. Telecommunication provides several new ways to communicate via computers.

INTERNET AND SOCIETY

Internet is an important and fast tool of exchanging information from national level to international level. With the passage of time more and more countries joined Internet to share its rich resources. Today, Internet is a **global web** of nearly 1 million computer networks, at least 50 million computers and 200 million people around the world. Internet hosts are connected to the Internet round the clock. Thousands more join this global network each month.

Internet provides many services to millions of people. Through Internet you can access thousands of databases and talk with experts worldwide on any subject. You can find jobs, get solution of your technical problems, sell products and conduct research. Internet has now become the need of most of us because of the vast array of information and activities possible with it.

1.8 APPLICATIONS OF COMPUTER

As computers have acquired more features, they have acquired more uses as well. Computers also have wide variety of uses in homes. Computer today can quote you the latest prices in the stock market. It can turn on the light of your living rooms, open door for you or teach you any language. Computers also offer you a variety of games and unlike most human partners they always want to play. Computers guide astronauts; maintain bank accounts, help engineers to design bridges and aeroplanes and control industrial units etc. Computers have radically changed the world of business. Computers can store every type of information and recall it almost instantly for use.

APPLICATIONS IN BUSINESS

The use of computers in business has been increased tremendously during the last two decades. Computers are used to record stocks of raw materials as well as finished products, making customer's bill, analyzing sales of various products and customer's trend, calculating employee's pay, commission and taxes and help to perform hundreds of other functions related to accounts, stocks, employees, customers, Importers, exporters, banks, stock exchanges, income tax etc.

BANKING APPLICATIONS

Banks are the largest users of computers. Computers are used for day-to-day processing of customers' accounts and payment of cheques, drafts etc. Computers inform when an installment of loan will be due. At some places customers can make purchases at stores using credit cards. The cards are inserted into a card reader at the store; it registers an automatic credit to the merchant's account at the bank, and makes an automatic debit from the customer's account.

STOCK EXCHANGE APPLICATIONS

Rapidly growing volume of the stock market has proportionally increased the paperwork problems. Traders had to wait to allow stock-market brokers to update before they may open business the following day. Computers can perform jobs efficiently with greater accuracy than it was done manually in the past.

RETAILING APPLICATIONS

In the commercial sector computers are used to write sale reports, financial statements, products cost, volume and profit analysis, keep a record of stolen, lost and damaged items. The black and white bar-code printed on most of the products in supermarkets is called Universal Product Code (UPC) shown in Figure 1.15. UPC is the key to computer-controlled supermarket check-out. The computer uses UPC to determine product's number, name, description and price.

Figure 1.15 Universal Product Codes in the form of series of vertical black bars and numbers are printed on supermarket products. At the check- out point, bar code reader is passed over UPC printed on the products. Bar code reader reads the information and sends it to the computer.

The customer has an accurate detailed receipt at the check-out cou terminal. While the computer is calculating and printing price; it is also upda the store's inventory list; warning the manager about an item going to be sl The information so obtained can also be used by marketing experts.

APPLICATION IN EDUCATION

The ability to use computers is a basic and necessary to a person's for education as reading, writing and arithmetic. There are number of method which educational institutions can use computers to educate the students.

COMPUTER ASSISTED INSTRUCTIONS (CAI)

It does not involve teaching about computers but rather using compu as an aid in teaching a subject. CAI is a system of individualized instruction uses a program presented by a computer as a learning medium. There are f major types of CAI systems:

- Drill and Practice
- Tutorial
- Dialog
- Testing

COMPUTER MANAGED INSTRUCTIONS (CMI)

In a computer-managed instruction (CMI) system, instead of teaching students directly, the computer assigns a student to read certain book, lis certain tape, attend certain lecture and so on. On completing the assignment, t student returns to the computer for testing and further assignment.

COMPUTER-BASED SIMULATIONS

To simulate is to copy the behavior of one system with a differe dissimilar system. Thus a computer can be programmed to behave like some oth system. Simulation is used when direct experimentation is impossible undesirable or uneconomical or immoral or simply too slow.

COMPUTER AIDED PROBLEM SOLVING

Students can use computers as problem solving tools. In **Computer-aide problem solving**, students explore and organize material from a course by usin computer as an aid to problem solving.

COMPUTER AS A SUBJECT OF INSTRUCTION

Today various types of computer science diploma and degree coursed ar offered by universities and colleges. During the past few years computer scienc have also been introduced at SSC and HSSC levels throughout the country Government is taking serious measure to promote computer literacy in th country. Large numbers of government and private institutions are engaged in computer education.

APPLICATIONS IN MEDICAL FIELD

Although computer has been introduced quite recently in the medical field, yet it finds numerous applications in this field. It is used for basic tasks such as keeping track of patient's appointments, diagnosis and treatment procedures. Imagine a medical information system that stores the case histories of millions of patients and made it available in seconds. The computer thus acts as a sort of medical consultant who can help a doctor to diagnose a disease and to prescribe best suitable treatment. The largest use of computers in the hospital is in the hospital administration in addition to patient's health care and medical research.

APPLICATIONS IN SCIENCE

Computer finds vast applications in the field of science and research such as in physics, chemistry, mathematics, Biological sciences, medical science, Agricultural science, Computer science itself, Nuclear Physics etc. In fact today it is impossible to carry out scientific researches without computers.

PRODUCT DESIGN AND MANUFACTURING

Companies that are manufacturing products have found new ways to use computers in every aspect of product development from the design stage to the manufacturing stage. **Computer-aided design** software **(CAD)** provides ways to develop a representation of the product and to test it in a variety of simulated environments.

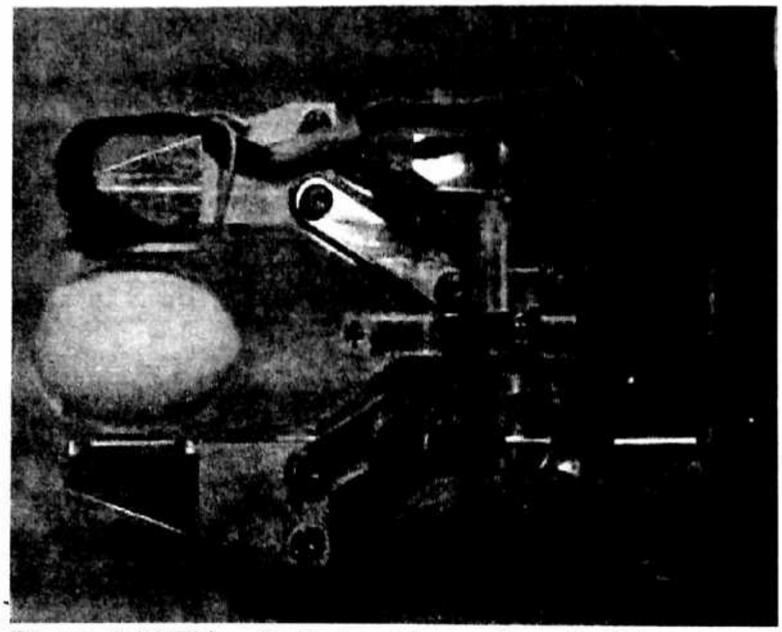


Figure 1.16 This robotic hand is capable of picking up and holding an egg without breaking it.

1.9 INTRODUCTION TO PROGRAMMING LANGUAGES

To perform a job, a computer needs instructions from the users. The instructions given to a computer to perform certain task is called a program. In other words programming is a way to communicate with computer.

We must realize that a computer does not understand any human language directly without some form of translation. A electronic digital computer uses binary number system having digits 1 or 0. These 0's or 1's represent on or off

state of certain electronic switch within the computer. The grouping of 0's and 1 in a certain sequence is called the machine code or machine language that wo cause the computer to perform certain operation.

Machine languages are not easy to learn. Therefore English-l languages called programming languages are used. These are easy to understan Programming languages can be classified as:-

- Machine Languages
- Assembly Languages
- Symbolic Languages

MACHINE LANGUAGES (*Low-level Languages*)

The set of binary instruction codes which a computer understands direct is called a machine language. It is different for different computers. Machi language is the most fundamental language of the computer. It is also called t **low level language**. Data and instructions are first converted in the machi language before it is sent to the computer. Combination of bits is used in vario codes to represent different characters, such as:-

Character	Machine Code	Character	Machine Code
A	●●○○○○○●	1	●●●●○○○●
B	●●○○○○●○	2	●●●●○○●○
C	●●○○○○●●	3	●●●●○○●●
Y	●●●○●○○○	(	○●○●●●○●
Z	●●●○●○○●	+	○●○○●●●○

ASSEMBLY LANGUAGES (*Low-level Languages*)

Machine language is the basic language of the computer but it is ver difficult. Assembly languages unlike machine language consist of short symboli phrases understandable by people. These phrases are made up of alphanumeri symbols called **mnemonics**. Assembly language simplifies the programmer's jot and makes it easy to find the errors. The programs written in assembly language are efficient in their use, and run faster as they use fewer instructions.

A program written in assembly language must go to an **Assembler** Assembler is a **system software** that resides in a computer. It translates assembl language into machine language before it runs on a computer. Assembl language is also a low-level symbolic language because it is very close to th machine code of a computer than the language of a problem. In assembl language, all operation codes of the machine language are substituted by letter and symbols of the **mnemonic code** by the manufacturers. Given below are th mnemonic code assigned to few instructions set.

INSTRUCTION	OPERATION CODE	MNEMONIC CODE
Stop processing	000 000	HLT
Add the number in memory to the number in accumulator	000 010	ADD
Multiply number in memory to the number in accumulator	000 100	MUL
The contents of the accumulator are stored in the location given by address part of instruction	011 000	STO
Load the number at the address in memory into the accumulator	000 001	LDA

Assembly languages are different for different computers. For a given microprocessor a user can devise ones own assembly language. Assembly languages have the following advantages over machine languages:

- *Easier to understand.*
- *Easy to locate errors and to correct.*
- *Easy to modify.*

SYMBOLIC LANGUAGE (*High level Languages*)

The difficulty in understanding the machine language by the user was resolved by symbolic coding using simple English-like words which a user can understand more easily. The programs written in these languages are then translated into machine language. The idea led to the development of a large number of different symbolic programming languages which are easy to use and understand. These languages are more suitable than machine languages for human use and enable the programmer to write instructions easily. These symbolic languages are called **high level languages**. High-level languages are completely general in application. Some of the popular high level languages are Basic, Fortran, Cobol, Pascal, Ada, C, Java.

BASIC (Beginner's All-purpose Symbolic Instruction Code)

BASIC was developed in 1964. It is sufficiently close to English and is one of the most widely used and easy-to-learn high level language. Even a person with little or no knowledge of computer programming can learn it quickly. BASIC can be used for both business and scientific applications.

FORTRAN (FORmula TRANslation)

FORTRAN was developed in 1957 for IBM computers. It was designed to solve mathematical, scientific and engineering problems. FORTRAN was one of the earliest languages to introduce the concept of modular programming. FORTRAN is a standard high level language. It has been revised many times. Its latest version is FORTRAN 90.

COBOL (Common Business Oriented Language)

COBOL was developed in 1959. It is an internationally accepted high level language developed for general, commercial and business purpose. COBOL was the first high level computer language suitable for handling large amount of data related to pay roll, credit and debit accounts, inventory and variety of other business applications.

PASCAL

PASCAL is a programming language named after the French mathematician and scientist Blaise Pascal. It was developed during the early 70's. Pascal is a highly structured programming language which is extremely popular in computer science.

ADA

ADA is a programming language for real time developed in 1980 for use in military applications. ADA is a high level structured programming language named after one of the first programmer Augusta Ada Byron.

C

C is the full name of a programming language developed in 1974 by Brian Karnighan and Dennis Ritchei. It is a favorite language of programmers for writing operating system. Programs written in C produce fast and executable code. C is also a very powerful language. With C, you can make a computer do just anything it is possible for a computer to do.

C++

C++ was developed by Bjarne Stroustrup in the early 1980s. Like C, C++ is an extremely powerful and efficient language. But C++ is even more difficult to learn than C. because C++ is a superset of C, learning C++ means learning everything about C, and then learning about object-oriented programming and its implementation with C++

GENERATIONS OF COMPUTER LANGUAGES

Programming languages are sometimes discussed in terms of generations. Each successive generation is thought to contain languages that is easier to use and more powerful than those in the previous generation. Machine languages are **first generation languages**, and assembly languages are considered **second generation languages**. The higher-level language began with third generations. Third level languages are portable unlike assembly languages. Program in these languages can be compiled to run on multiple CPUs. These languages include FORTRAN, COBOL, BASIC, PASCAL, C, C++.

Fourth generation languages (4GLs) are mostly special purpose programming languages that are easier to use than third-generation languages

Fourth generation languages (4GLs) also use computer power to convert them into machine code. With 4GLs, programmer can create applications quickly without specifying each detail.

VISUAL BASIC

One of the 4GL which is not a special purpose language is Visual-Basic. Visual Basic supports **object-oriented** features and methods. Visual programming takes object-oriented programming to the next level, replacing the text-based instructions with symbolic icons, each of which represents an object or common program function. For example, to place a box on a form, Visual Basic programmers simply drag the box form a toolbox onto the form at an appropriate place and resize it. The necessary code for the placement and size of the box is written automatically. Programmers find it easy to write programs quickly with relatively little effort.

JAVA

Java is an object-oriented programming language developed in 1991. it was originally developed for use in set-top boxes, transitioned to the World Wide Web in 1994. Modeled after C++, the Java language was designed to be small, simple, and portable across platforms and operating systems, both at the source and at the binary level. Java was written as a full-fledged general-purpose programming language in which you can accomplish the same sorts of tasks as in other programming languages, such as C or C++.

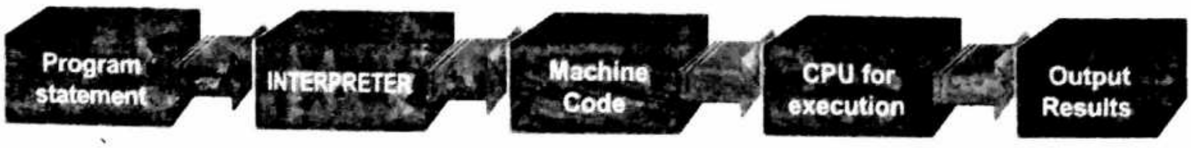
1.10 LANGUAGE PROCESSORS

A language processor is a **translating-software** that enables a computer to respond the user's instructions. Since it is very difficult to write a program in machine language that computer can understand. Hence it becomes necessary to convert user's instructions into machine code by using a language processor. There are two types of language processor:-

i) Compilers ii) Interpreters

INTERPRETERS

An interpreter translates a program written in a high-level language into its equivalent machine code. It translates the statements in high-level language one by one and executes it before translating the next statement of the source program.



The advantage of an interpreter is its fast response to changes in the source program. Interpreters are easy to write and also do not require large computer

memory. However it is a time consuming processor because each statement from the source program must be translated and executed, before translating the next source statement and thus wastes a lot of time and effort.

COMPILERS

A compiler is complex system software that automatically converts a program written in some high-level language into an equivalent **low-level machine language**. The computer or the language processor converts the entire program into machine code before execution. A program written by a programmer in a language other than machine language is called a **source program**. The output from a compiler or an assembler, which consists of machine language instructions, is called the **object program**.

A compiler can translate only that source program that has been written in a language for which it is designed. For example a FORTRAN compiler is only capable to translate a source program written in FORTRAN. It differs from the interpreter which translates one statement at a time , execute it before translating the next statement.

SUMMARY

HISTORY AND DEVELOPMENT OF COMPUTER

Man is always finding ways to ease his physical and mental labour. The Computer has evolved through centuries. From the time the man first started using arithmetic, he had been inventing devices to handle numbers. One of the first recognized calculating devices was the Abacus developed around 3000BC.

In 1614 John Napier invented logarithms and a device called Napier's bones or rods. The idea gave birth to another very useful calculating device called **Slide Rule**. In 1642 Pascal developed a machine known as Pascaline that could add

and subtract. In 1672 Leibnitz developed a calculating machine that could add, subtract, multiply and divide. In 1790 Joseph Maria Jacquard built a loom introducing the concept of programming using punched cards.

In 1786 J. H. Muller proposed a calculating machine called a difference engine. Charles Babbage in 1812 began to develop a difference engine. After spending 18 years, he started working on a new machine called the Analytical Engine. The machine had to be programmable and fully automatic. Babbage's analytical engine could not be completed but he laid the foundation stone for the development of modern digital computers. He is called the father of modern digital computer. In 1887 Dr. Herman Hollerith designed a Tabulating machine. His Tabulating Machine could read and sorted out data from punched cards.

Mechanical calculators were heavy, large and slow. The first computer that used electromagnetic relays was the **complex calculator.** It was the first machine that used **binary number system**. In 1944 Dr. Howard Aiken designed **Automatic Sequence Controlled Calculator** (ASCC) or **Mark-I.**

GENERATIONS OF COMPUTER

The development of electronic computers can be divided into generations, depending upon the technologies used. The generations of computers are:-

- **First generation** (Vacuum tube1906) 1942 - 1959
- **Second generation** (Transistors1947) 1959 -1965
- **Third generation** (Integrated circuits IC's 1965) 1965 - 1973
- **Fourth generation** (Microprocessors 1971) Since 1972
- **Fifth generation** (Artificial Intelligence) In progress since1980s

TYPES OF COMPUTER

Computers can be divided into three types. This division depends upon the design and working of a computer. These three types of computers are Analog Computers, Digital Computers, Hybrid Computers

CLASSIFICATION OF COMPUTERS

Computers can be classified into the following four main groups according to their sizes processing power, speed, size of main memory and other capabilities: Supercomputers, Mainframe, Minicomputer and Microcomputer.

COMPUTERS AND SOCIETY

Computers are becoming more and more involved day by day in our lives. We are using computers in education and health, in trade and industry, in defence and agriculture, in research and in entertainments.

But at the same time we are also facing problems arising from the use of computers. These are unemployment due to automation, wastage of time and energy in useless computer activities, data security, personal privacy and computer crimes.

INTERNET AND SOCIETY

Internet is an important and fast tool of exchanging information. It ha changed the entire world into a global village.

APPLICATIONS OF COMPUTER

Computers guide astronauts; maintain bank accounts, help engineers design bridges and aeroplanes and control/monitor industrial units etc. Compute have radically changed the pattern of trade. Computers can store information a recall it almost instantly for use. It finds applications in every field of life.

LANGUAGE PROCESSORS

A language processor is a translational software that converts user instructions into machine code. There are two types of language processor:-

i) Compilers   ii) Interpreters

MACHINE LANGUAGES

The set of binary instruction codes which a computer understands directl is called a machine language. It is also called the low level language.

ASSEMBLY LANGUAGES

Assembly languages unlike machine language consist of short symbol phrases understandable by people. Assembly language is also a low-lev symbolic language. In assembly language, all operation codes of the machin language are substituted by letters and symbols called **mnemonic code**.

SYMBOLIC LANGUAGE

In symbolic language, simple English-like words are used. A user ca understand them more easily. The programs written in these languages are the translated into machine language. These symbolic languages are called high leve languages. Some high level languages are Basic, Pascal, Ada, C, and Java.

INTERPRETERS

Interpreters are the type of translators that convert each statement of program written in a high-level language into machine code and execute it befor translating the next statement of a program. Interpreters are easy to write and als do not require large computer memory.

COMPILERS

A compiler is a system software that automatically converts a prograr written in some high-level language into an equivalent low-level machin language. It differs from the interpreter which translates a statement, execute i before translating the next statement.

EXERCISES

1.01 Complete the following statements.

i) A modern computer is an electronic ________________ device.

ii) One of the first recognized calculating devices was the__________.

iii) ___________is called the father of modern digital computer.

iv) Pentium II is a ___________generation computer.

v) ___________is a system of worldwide computer network for exchanging information.

vi) UPC stands for______________________.

vii) ______________language consists of short phrases of alphanumeric symbols called mnemonics.

viii) BASIC stands for____.

ix) Interpreter translates a symbolic language into________________.

x) In OOP text-based instructions are replaced by________________.

1.02 Tick (✓) the following statements either True or False.

i) Pascal used Napier's idea of logarithm in Pascaline. True/False

ii) Fourth generation computers possess the ability of AI. True/False

iii) Analog computers are faster than digital computer. True/False

iv) Assembly language is a HLL. True/False

v) C is the most favorite language for writing Operating System. True/False

1.03 Encircle one choice A, B, C or D in each of the Multiple choice questions.

i) The first computer that used Vacuum tubes was:-
(A) Mark I (B) ENIAC (C) EDVAC (D) UNIVAC

ii) Vacuum tube was invented in:-
(A) 1899 (B) 1906 (C) 1916 (D) 1926

iii) Which of the following is called the first generation computer?
(A) Abacus (B) Pentium I (C) Mark I (D) UNIVAC

iv) Transistor was invented in:-
(A) 1943 (B) 1947 (C) 1950 (D) 1952

v) A translator is a:-
(A) System software (B) Low-Level Language
(C) Translating machine (D) Application software

1.04 Match the items given in Column I with those given in Column II

	Column I		Column II
i)	ADA	a)	Fifth Generation
ii)	Operation System	b)	4 GL
iii)	Artificial Intelligence	c)	CAD
iv)	Visual Basic	d)	CAI
v)	Transistor	e)	LLL
vi)	Microprocessor	f)	Translator
vii)	Software	g)	Real Time
viii)	Machine Language	h)	C
ix)	Teaching/Learning	i)	VLSI
x)	Compiler	j)	Second Generation

1.05 Describe various units of Babbage's analytical Engine.

1.06 Name various parts used in the first electromechanical calculator.

1.07 Describe basic features of three generations of computers.

1.08. Name one model of each generation of computer.

1.09. Describe the basic difference between an analog and a digital computer.

1.10. Describe some of the features of :-

a) A Microcomputer b) A Supercomputer

1.11. What is Internet? Describe its important in society.

1.12 Write five advantages and five disadvantages of computer in our life.

1.13 List some reasons for the use of computers in banks.

1.14. Describe some features of a computerized supermarket check-out system.

1.15. What is UPC?

1.16. What is meant by CAI?

1.17. Briefly define computer science education.

1.18. List some reasons for using computers in hospitals.

1.19. What is meant by CMI?

1.20 Define a computer program.

1.21. Why machine and assembly languages are called low level languages?

1.22. What do you understand by Mnemonic code? Write down some of the Mnemonic codes along with the operations assigned to these codes.

1.23. What are high level programming languages? How they differ from Assembly language?

1.24 Write short notes on the following:-

(a) BASIC (b) PASCAL (c) FORTRAN

1.25. What is a compiler? How it differ from an interpreter?

2

COMPUTER COMPONENTS

A computer is an electronic data processing device. It reads data, processes it and produces results accurately at a very high speed. A Computer along with a number of units attached to it (such as keyboard, monitor, disk drives etc.) is known as a **computer system**.

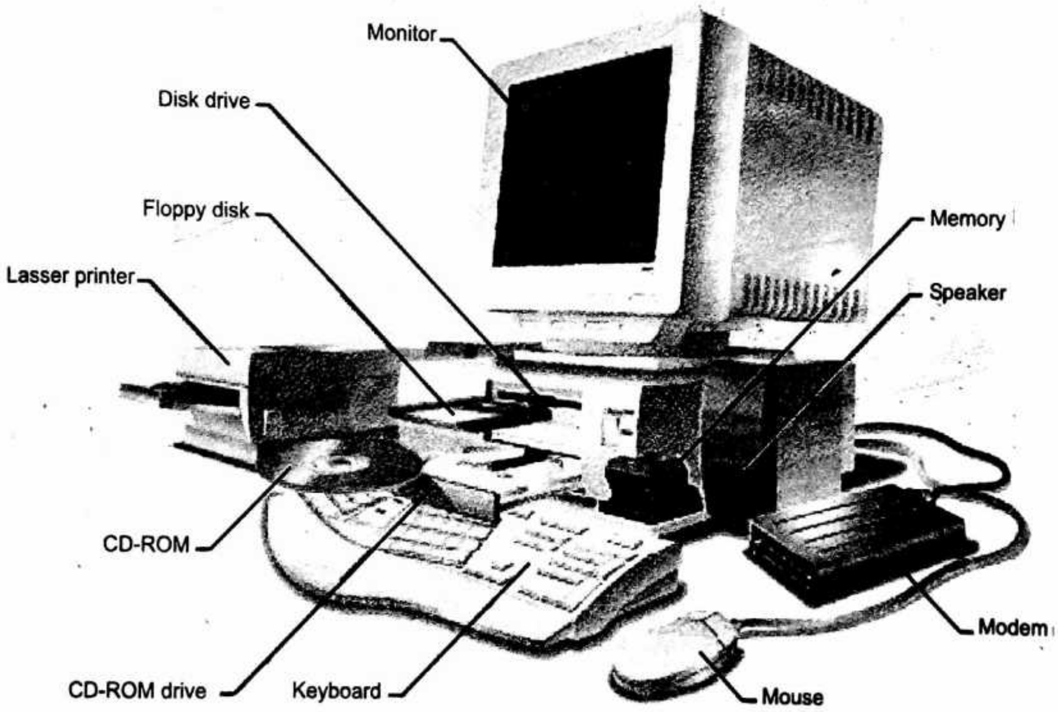


Figure 2.1 A computer system

2.1 DIVISION OF A COMPUTER SYSTEM

Computer systems have two major divisions:-

- **Computer hardware.**
- **Computer software.**

COMPUTER HARDWARE

Computer hardware are the physical components which are installed in a computer box as well as the devices connected with it such as key board, monitor

mouse etc. as shown in Figure 2.1. **Main unit** or the **system unit** consists of control unit, processing unit and memory units.

a) System Units

(i) Desktop type

(ii) Minitower type

b) Input Units

(i) Keyboard

(ii) Mouse

c) Output Units

(i) Printer

(ii) Monitor.

Figure 2.2 Computer Hardware (a) System units (b) Input devices (c) Output devices

The devices that are attached with a system unit under the control of CPU are called as the peripheral devices. Some common peripheral devices are keyboard, mouse, monitor, printer and disk drives. These devices help to communicate data/information with the system unit. Generally a computer system consists of the following three major units:-

- **System unit**
- **Input units** (A keyboard, mouse etc.) and
- **Output units** (A monitor, printer etc.)

A computer system performs the following functions:-

- *Receive data and instructions for the system.*
- *Stores the instructions and data to be used for processing.*
- *Executes the instructions to process the data.*
- *Transfer the results in a form usable by humans or other machines.*

COMPUTER SOFTWARE

The techniques or methods necessary to get the **hardware** to work are known as **Computer Software**. In other words, **Computer Software is the know-how techniques in the form of a set of instructions that tells the computer what to do**. Computer software can further be divided into:-

- System Software
- Application Software

2.2 COMPUTER ORGANIZATION

An electronic digital computer is a vast assembly of **simple logic gates** and **memory cells.** All these components are organized to perform arithmetic and logic operations at high speed. The design and performance of various systems vary widely however they follow the same general plan as shown in Figure 2.3.

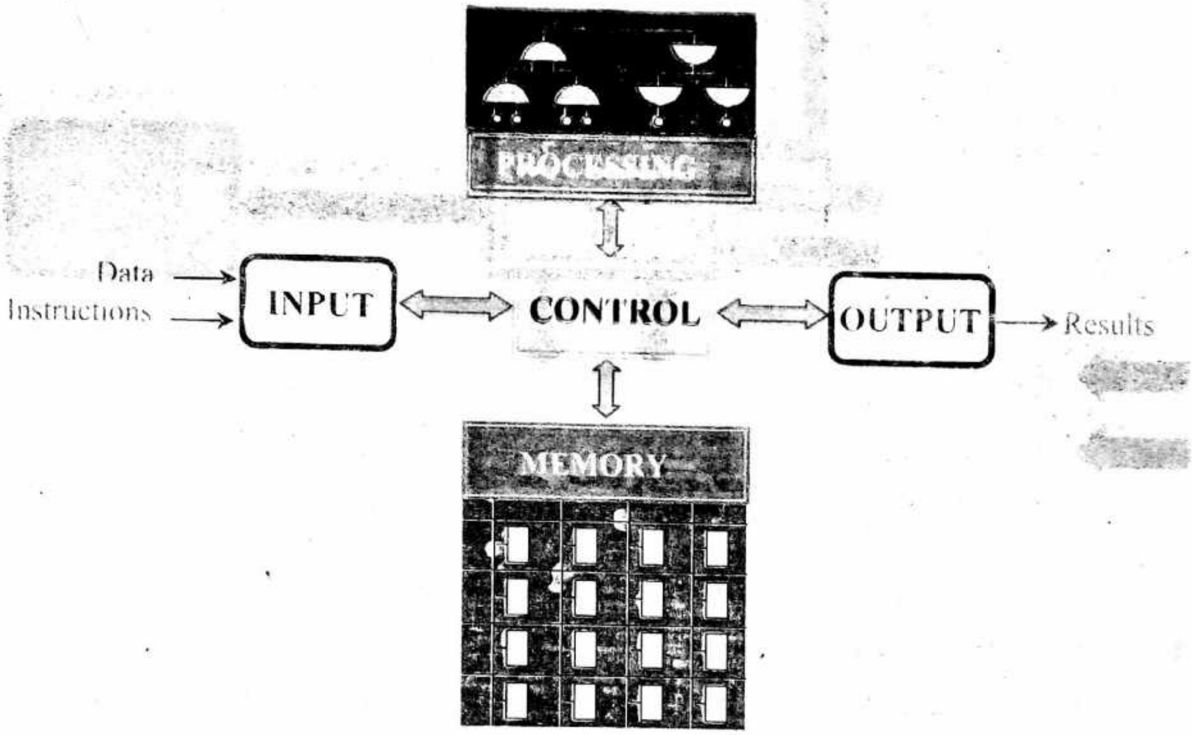


Figure 2.3 Functional components of a general-purpose Digital Computer

THE SYSTEM UNIT

The system unit is the main unit of a computer system. It allows various parts of the computer system to work together. A system unit consists of :-

- *The Central Processing Unit (CPU)*
- *Semiconductor memory*
- *Magnetic memory and Disk drives*
- *Adapters and Connectors*

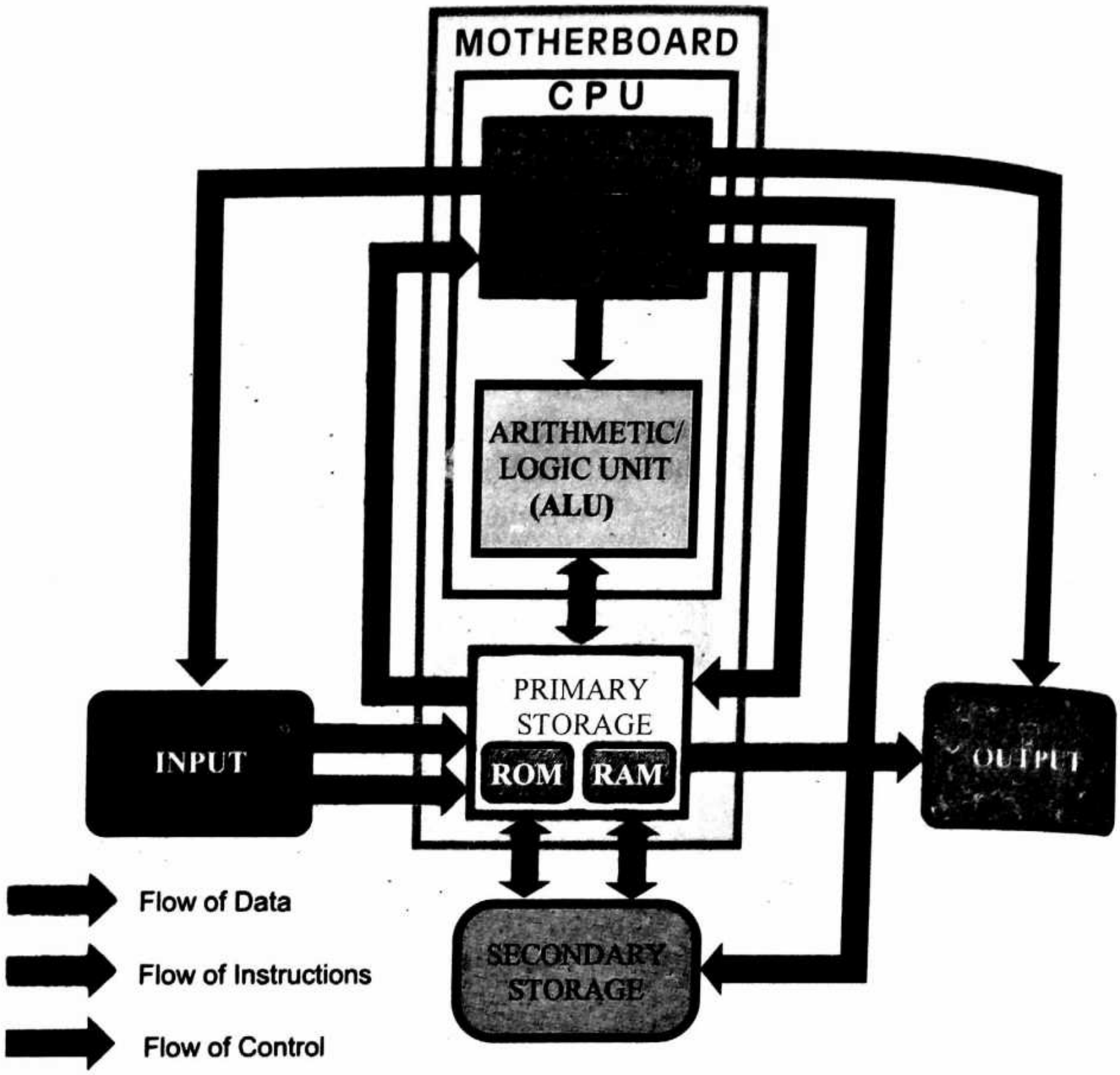


Figure 2.4 Block diagram of a Computer System with sub-units of CPU.

2.3 CENTRAL PROCESSING UNIT (CPU)

The central processing unit (CPU) and memory unit of a computer are located inside the system unit. The central processing unit (CPU) is the brain of a computer. It controls the entire computer system and executes the instructions given to it. The CPU processes the data as instructed and sends the results to an output unit. A CPU chip or microprocessor consists of the following two units:-

- **Control Unit (CU)**
- **Arithmetic and Logic Unit (ALU)**

Note that the **CPU's control (CU)** uses a **clock** and a **system bus**. The clock sends out **electric pulses** at a set rate and **regulates the timing** of all **system operations**. The system bus is a **communication system** that provides a **path** for the **flow of instructions, data,** and **control commands.**

CONTROL UNIT (C.U)

The Control Unit is the most important part of the CPU. It controls and coordinates the activities of all the units of a computer system. It performs this function by issuing necessary commands to respective units of the computer.

Both the CU and the ALU require a small amount of temporary storage. These small temporary storages or internal memory units hold the instructions and data currently being processed. These small memory units are called **registers**. The control unit **fetches** an instruction from RAM and stores it in an **instruction register** and moves it to one of the ALU registers.

ARITHMETIC AND LOGIC UNIT (ALU)

Arithmetic and logic unit (**ALU**) contains a number of **adder and logic gates** in addition to various types of **registers**. ALU performs all the processing in a CPU. It also carries out arithmetic operations like addition, subtraction as well as logical operations. Registers such as address registers, accumulators, instruction registers, program counter, etc. help the computer to process data.

Let us understand this by the **addition** of two numbers **A** and **B** stored in the main memory of the computer. The Control Unit **interprets** the instruction and **transfers** the number **A** from main memory to the **Accumulator**. The control unit then **transfers** the number **B** to some other **register** available. Under the **instructions** of the Control Unit, these numbers are **added** by the **adder circuits** and the **result** is **stored** in the **accumulator**. From the accumulator it may be transferred back to the main memory, from where it may be sent to other units when required.

2.4 COMPUTER MEMORY

CPU contains registers which are small storage areas and A CPU needs much larger space in mega-bytes as compared to registers that can hold a few bytes at a time. Computers are equipped with storage units also called memory units linked with the processor. The computer memory can hold programs and the data for current and future use. These storage devices can be classified as:-

- Main memory (*Primary storage*)
- Secondary memory (*Secondary storage*)

MAIN MEMORY

Main memory of a computer is its internal memory also called the primary memory. Generally the main memory is on semiconductor chips as shown in Figure 2.5. These memory chips are connected with the CPU on the motherboard. Main memory is very fast as it is directly accessible by the CPU.

Different types of memory chips are available such as RAM (Random Access Memory), ROM (Read Only Memory), PROM (Programmable Read Only Memory) and EPROM (Erasable and programmable Read Only Memory etc.

The capacity of a storage device is generally expressed in **bytes**. One byte is a group of eight **bits**. A **Bit** is a smallest memory unit that represents a **binary digit**. A bit has value either **0** or **1.** Eight bits or one byte is used to represent a character. Thus 8000 bytes memory means a storage capacity of 8000 characters.

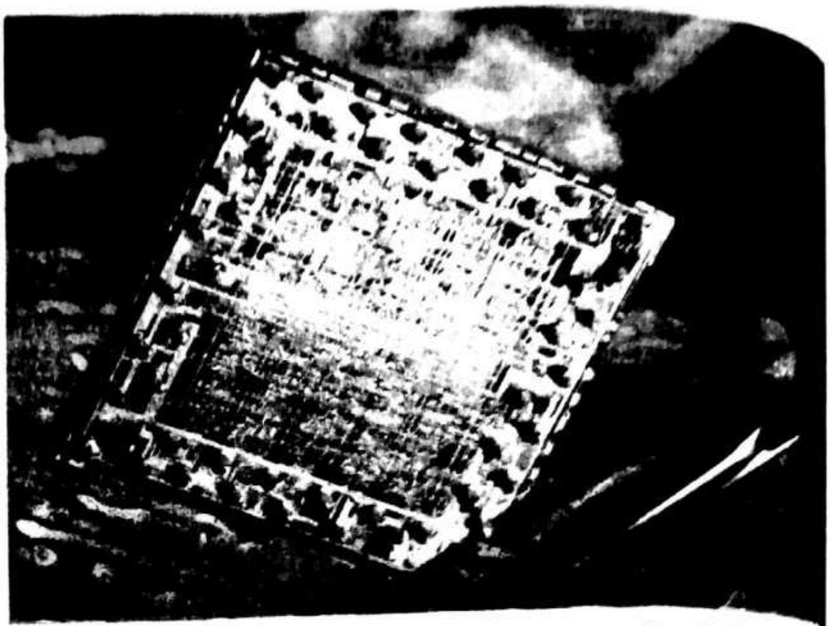


Figure 2.5 This tiny chip has a storage capacity of 65536 bits of information(or about 8000 characters.)
For comparison, the chip is shown on a needle

READ ONLY MEMORY (ROM)

Read Only Memory (ROM) is a permanent memory of the computer. It handles many important tasks. One of these tasks is to provide instructions to CPU during start-up or booting process whenever the computer is **turned on**.

Another important role of ROM is that of **character generator**. When a key is depressed, it sends a signal to ROM which in turn supplies a character to be displayed on the screen. ROM is a **nonvolatile memory** i.e. the instructions stored on it cannot be changed or deleted when the computer is **turned off**.

RANDOM ACCESS MEMORY (RAM)

Random Access Memory (RAM) is major part of the main memory. Any part of RAM is directly accessible to **read/write** data or instructions on it. It is a fast memory but not permanent like ROM as it loses all of its data and instructions as soon as the computer is **turned off**. Thus RAM is also called a **volatile memory**

SECONDARY MEMORY (Backing Storage)

Secondary memory also called as **backing storage** is used to supplement the capacity of main memory. Secondary storages such as Floppy disks, hard disks and CDs can store a bulk of data/information. The data in secondary storage is first brought into the main memory before processing.

2.5 BUSES

In a computer system large amount of data and instructions have to move from one unit to the other. This requires a **system bus,** which is a set of parallel

wires. It provides an electrical path among Control unit, ALU, Memory and other hardware parts on the motherboard or attached circuit boards. Addresses, data and control commands all flow along the system bus through separate paths called the address bus, the data bus and the control bus respectively.

THE DATA BUS

The data bus is an electrical path that connects the CPU, memory, and the other hardware devices on the motherboard. The number of wires in the bus affects the speed at which data can travel between components. Since each wire can transfer one bit at a time therefore an 8-wires bus can move 8 bits or one byte at a time. A 16-bit bus can transfer 2 bytes at a time and so on. Thus the speed of the computer at which it receives and sends data between various units depends upon the **bus width.** Bus width is the number bits that it can carry at a time.

THE ADDRESS BUS

The second bus that is found in every microcomputer is the address bus. The address bus that carries the memory addresses is a set of wires similar to the data bus. An address bus connects the CPU with the memory.

The reason that the address bus is important is that the number of lines in it tells the maximum number of memory addresses. For example, 8 bit data is enough to represent $2^8 = 256$ different values. Thus a CPU with 8-bit address bus could address only 256 bytes of memory. Similarly a CPU with 16-bit address bus can address $2^{16} = 65,536$ bytes = 64 Kbytes of memory and a CPU with 32-bit address bus can address $2^{32} = 4,294,967,296$ bytes = 4 GB of memory and so on.

THE CONTROL BUS

The Control Unit controls the activities of all other units of the system. It sends out control commands to all the units through control bus. Control unit directs the transfer of data from main memory to arithmetic and logic unit (ALU) for processing. The control bus provides a two-way transfer of data, instructions and the results between the main memory and secondary storage.

2.6 THE PORTS

A system unit can work only when linked with Input/Output devices such as a keyboard and a monitor. Each I/O device has a cable with a plug having several pins for plugging into a socket called port. A port also called an interface is generally provided at the back of a Computer. A port provides a direct link with the computer's common electrical bus.

Most computers have several types of ports, each with different capabilities and uses. External peripheral devices can be linked to the system unit

through a **serial** or **parallel port**. Most of the PCs come with I/O cards that provide upto four serial ports, two parallel ports plus a dedicated keyboard port.

SERIAL PORTS

A serial port provides a connection for plugging a peripheral device with one data line that carry one bit at a time and additional lines to carry control signals. A serial port connects your computer to a device such as a modem, which requires two-way data transmission, or to a mouse, which requires only one-way data transmission. A serial port can have 9-pins or 25-pins. In Personal computers serial ports are generally designated as COM1, COM2 and COM3 etc.

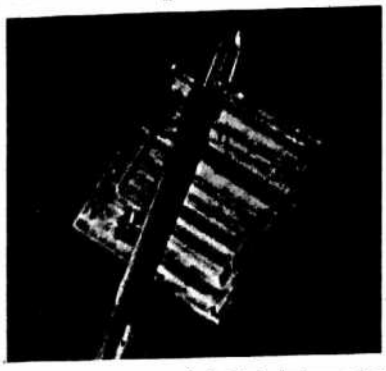
PARALLEL PORTS

A parallel port provides a connection for transmitting fast flow of large amount of data. Parallel ports allow 8 bits data at a time over a cable with eight separate data lines. Parallel ports can handle larger volume of data at a much faster rate as compared to serial ports. Parallel transmission is used to send data to the printer.

The cable that connects two parallel ports contains 25 wires, 8 wires carry data and the remaining wires carry control signals. The parallel port is the simplest link with peripherals which are kept near the main computer. In Personal computers parallel ports are generally designated as LPT1, LPT2 and LPT3 etc.

2.7 MICROPROCESSORS

A microprocessor is Very **Large-Scale Integrated** circuits **(VLSI)** developed on a single semiconductor chip. In spite of its small size a the chip contains thousands of electronic components. It is the central processing unit of a computer without having a significant amount of memory. Microprocessors such as shown in Figure 2.6 (a) are used in calculators, cameras microwave ovens, washing machines point-of-sale terminals, medical equipments and many other devices. New microprocessors used in PCs Figure 2.6 (b) contains millions of transistors and other components.



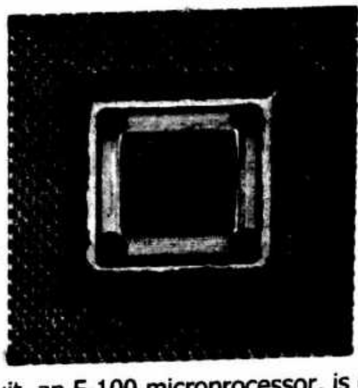


Figure 2.6 Microprocessors (a) This integrated circuit, an F-100 microprocessor, is only 0.6 cm square and is small enough to pass through the eye of a needle.**(b)**A Pentium microprocessor with its pin connections

2.8 The Motherboard

A complete digital system including microprocessor (**μP**), program memory (ROM), data memory (RAM), with other electrical components and input/output ports (I/O) housed in a system unit is called a microcomputer.

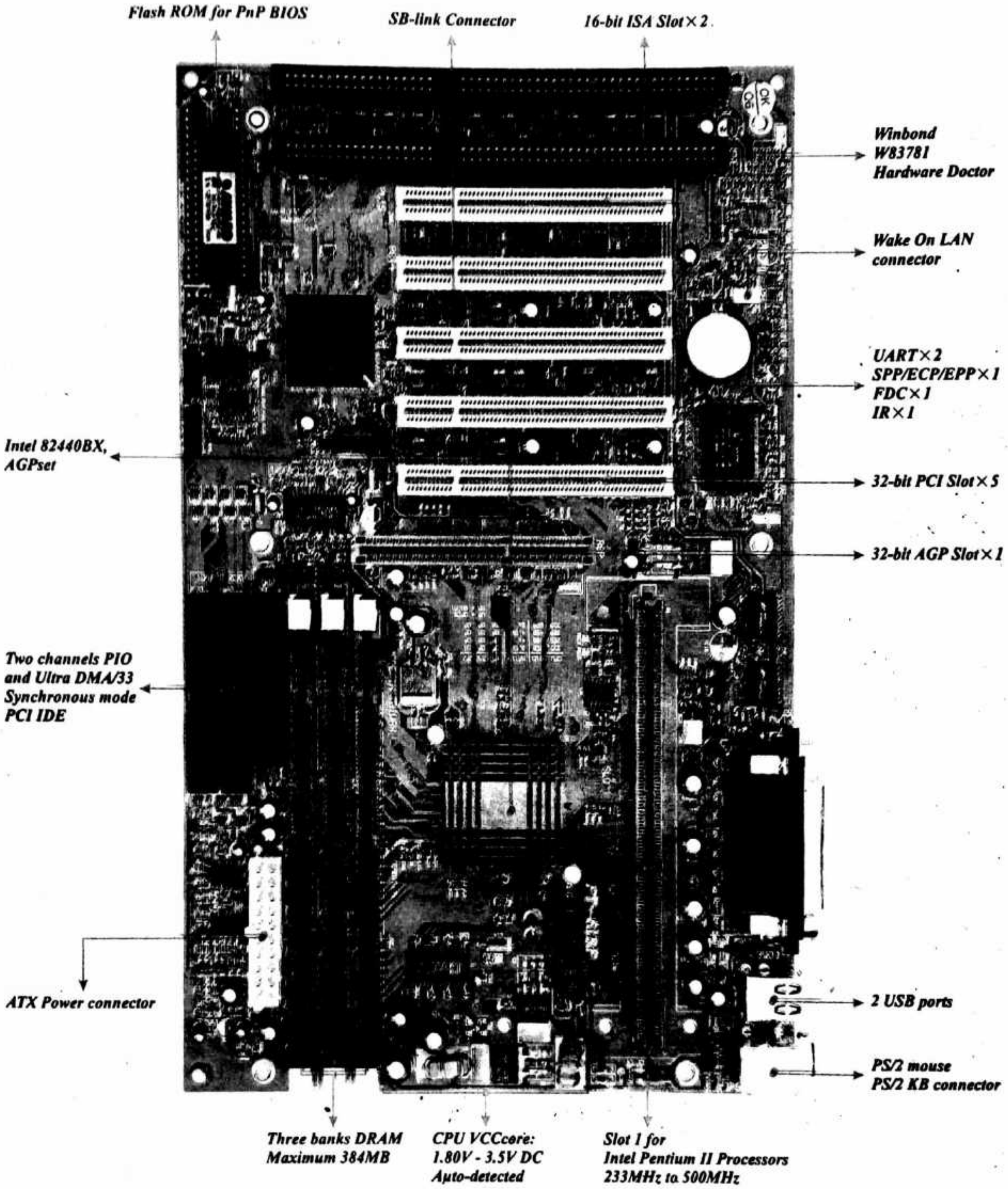


Figure 2.7 A motherboard containing a microprocessor and expansion slots

In microcomputers CPU-board is called the main board or motherboard. Figure 2.7 is a photograph of a Motherboard showing a microprocessor with basic

circuitry and other components such integrated circuits, transistors, dio resistors etc. Integrated circuits also called chips alongwith the microproce chip are required in a system unit to store and process data. Data moves from chip to another along conducting lines called a bus.

SUMMARY

A modern computer is an electronic data processing machine. A Compu along with a number of units attached to it is known as a **computer system**.

DIVISION OF A COMPUTER SYSTEM

A Computer system has two major divisions called the Comput hardware and Computer software.

COMPUTER HARDWARE

Computer hardware is the physical components of the computer systen These are System unit Input units (such as keyboard) and Output units (such a monitor (video display screen)

COMPUTER SOFTWARE

The techniques or methods necessary to get the hardware to work ar known as Computer Software. Computer software can further be divided into: System Software and Application Software

CENTRAL PROCESSING UNIT (CPU)

The central processing unit (CPU) is the brain of a computer. It controls the entire computer system and executes the instructions given to it. It consists of Control Unit and Arithmetic & Logic Unit. It is located inside the system unit.

CONTROL UNIT (C.U)

The Control Unit is the most important part of the CPU. It controls and coordinates the activities of all the units of a computer system. It performs this function by issuing necessary commands to respective units of the computer.

ARITHMETIC AND LOGIC UNIT (ALU)

Arithmetic and logic unit performs all the processing in a CPU. It also carries out arithmetic and logical operations.

COMPUTER MEMORY

Computer memory can hold programs and the data for current and future use. It is mainly of two types called the Main memory and Secondary memory.

MAIN MEMORY

Main memory consists of RAM and ROM built on semiconductor chips. These are very fast and are directly accessible by the CPU. ROM is a permanent memory of the computer. It handles many important tasks such as booting. While RAM which is a major part of the main memory is a volatile memory and loses all of its data and instructions as soon as the computer is turned off.

SECONDARY MEMORY (Backing Storage)

Secondary memories such as floppies, hard disks and CD's are used to increase the capacity of main memory. They can store a bulk of data/information.

BUSES

A bus is a set of parallel wires. It provides an electrical path among Control unit, ALU, Memory and other hardware parts on the motherboard or attached circuit boards. Different types of buses are address bus, the data bus and the control bus to send addresses, data and control commands respectively.

THE PORTS

The receptacle at the back of a PC is called a port. A port provides a direct link for external peripheral devices such as keyboard, mouse, monitor, printer etc via cables with the computer's common electrical bus. Most computers have several types of ports such as serial port, parallel port and dedicated each with different capabilities and uses.

MICROPROCESSORS

A microprocessor is the central processing unit of a computer without having a significant amount of memory. It is a Very Large-Scale Integrated circuits (VLSI) developed on a single semiconductor chip. In spite of its small size a microprocessor chip contains thousands of electronic components. Microprocessors are also used in calculators, cameras microwave ovens, washing machines point-of-sale terminals, medical equipments and many other devices. In microcomputers CPU-board also called motherboard contains many chips for other components in addition to one or more microprocessor chips.

EXERCISES

2.01 Complete the following statements

i) A system unit of a computer consists of a _______ and_______.

ii) CPU stands for______________________________.

iii) ALU stands for______________________________.

iv) Internal memory unit that are the small temporary storage (computer are called_______________.

v) A 16 bit bus can transfer _____________ bytes at a time.

2.02 Tick (✓) the following statements either True or False.

i) The CPU of a computer is one of its input unit. True/F

ii) The CPU of a system processes the data as instructed and sends the results to an output unit. True/F

iii) The Control unit carry out logical computation. True/F

iv) Registers are the small storages the data that hold the Instructions and data currently being in process. True/F

v) One of the important function of RAM is to generate Characters. True/F

2.03 Encircle one choice A, B, C or D in each case.

i) which of the following are present in a system unit.
(A) I, II & III (B) II, & III (C) I, II& IV (D) I, II, III &
where (I) *Input unit*, (II) *CPU*, (III) *Memory unit and* (IV) *Output unit*

ii) RAM of a Computer is a:-
(A) Permanent storage (B) Secondary storage
(C) Primary storage (D) Backing storage

iii) The brain of an electronic computer is its:-
(A) CPU (B) CU (C) ALU (D) ROM

iv) An 8-bit address bus can address a memory of:-
(A) 1 Byte (B) 8 Bytes (C) 256 bits (D) 256 Bytes

v) Which of the following is not a semiconductor memory?
(A) ROM (B) RAM (C) PROM (D) None of the

2.04 Match the items given in Column I with those given in Column II

	Column I		Column II
i)	Hardware	a)	Booting Process
ii)	Software	b)	LPT
iii)	Serial Port	c)	Physical Components of a Computer system
iv)	Parallel Port	d)	Procedure necessary to run the Computer
v)	ROM	e)	COM

2.05 Differentiate between hardware and software of a computer.

2.06. Describe the main functions of CPU

2.07 Describe the working and function of ALU.

2.08 Discuss the importance of main memory. Explain the types of main memory.

2.09 Describe the function of control Unit.

2.10 What is meant by secondary storage? Mention some secondary storage devices.

2.11. What is meant by bus in computer system? How is Data bus differs from an address bus?

2.12. Describe the function of ports in a computer. How many types of ports are generally present in a computer system?

INPUT/OUTPUT DEVICES

We need to communicate with computer to send data and instructions in the computer and get the processed data back from it. This is done by t **input/output devices** or **I/O devices**. **Input devices** receive data and instructio from outside, and send them to CPU. The CPU processes the data as directe **Output devices** retrieve processed data from the computer's memory and prese it into a suitable form familiar for human or save it permanently for future us Thus without Input/Output devices, we cannot communicate with computer.

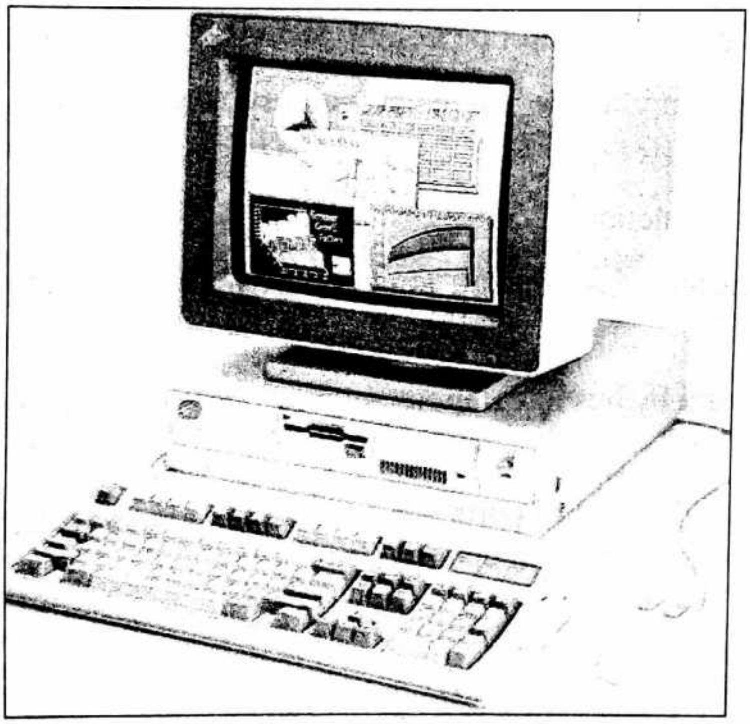


Figure 3.1 A personal computer system

3.1 INPUT DEVICES

Input devices are used not only to enter data into the computer but also the instructions or programs that tell the computer what to do with the given data. Some of the common input devices are keyboard, mouse, joystick, optical character reader, light pen, scanner, disk drive etc.

3.2 KEYBOARD

Keyboard is an important input device used to enter data and instructions into a computer. A standard arrangement of keys is the QWERTY arrangement. Popular keyboards have enhanced QWERTY arrangement designed for easy entry of numbers. Such keyboards have numeric keypad at the right of the standard keyboard. A keyboard is connected with the system unit through a dedicated keyboard port that contains signal as well as power lines. As soon as a key is depressed for a character a signal for that character is generated that passes directly to the main storage of the computer.

KEYBOARD STYLES

There are two basic styles of keyboard commonly used with micro computers. These are PC/XT-style keyboards and AT-style keyboards.

a)

b)

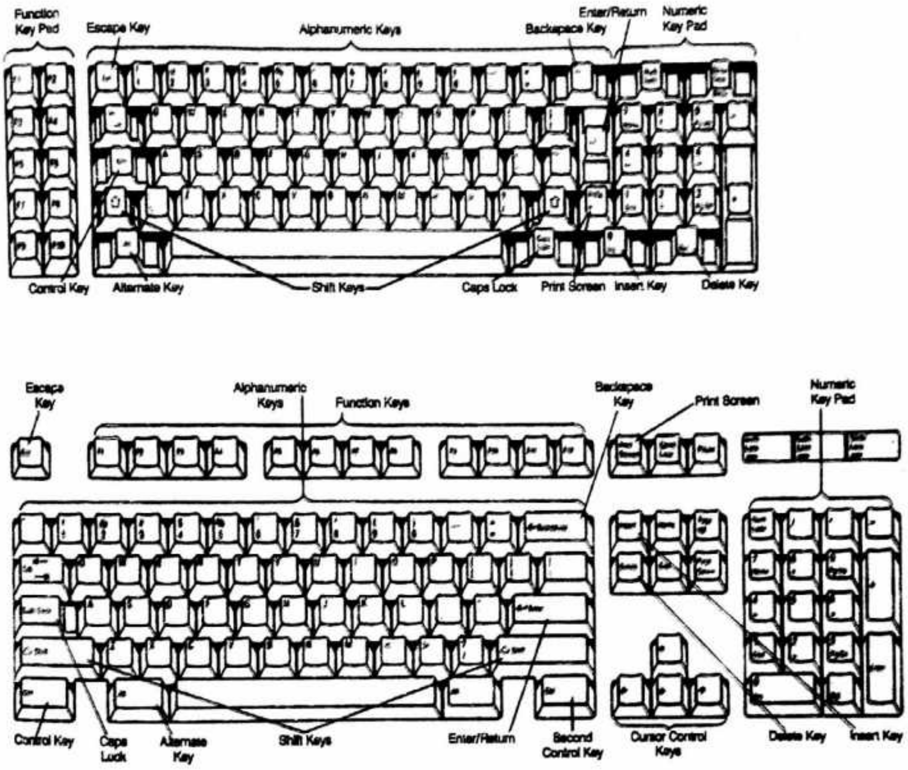


Figure 3.2 The two most common keyboards styles (a) PC/XT style (b) AT style

AT-style was developed later with a slightly different keyboard having 101 keys. Various manufacturers have developed keyboards using different styles with slight alteration having 84 to 108 keys.

DIVISION OF A KEYBOARD

A keyboard may be divided into four general areas:-

- Alphanumeric keypad.
- Function keypad.
- Numeric keypad.
- Screen Navigation & Editing keys.

ALPHANUMERIC KEYPAD

The position of various keys in the Alphanumeric Keypad is similar to those on a typical typewriter and is also called typewriter's area. as shown in Fig.3.2. Alphanumeric Keypad consists of:-

a) **Alphabet keypad** consists of keys A to Z having upper case and lower case letters.

b) **Number keys** consist of keys from 0 to 9.

c) **Punctuation keys, special character keys and space bar** for punctuation marks, special characters such as +) * & % $ # @ } etc.

NUMERIC KEYPAD

A numeric keypad is a set of keys at the right of the keyboard similar to those on a calculator. These are dual-purpose keys used for rapid entry of numeric data as well as for editing and cursor control.

FUNCTION KEYS

There are 10 or 12 function keys labelled as F1, F2, ... F10 or F1, F2, ... F12. These are located on the left or at the top of the keyboard. These are used to instruct the computer to perform certain commands. These keys provide short cuts for doing routine tasks on a computer.

Function keys perform different tasks in different **software packages.** The function keys can also be used in combinations with one or more other keys such as **SHIFT**, **CTRL**, **and ALT**. These combinations are the shortcuts to carry out certain commands. For example,

SHIFT + **F2**; copies the text (**Edit** menu)
CTRL + **F2**; chooses the print preview (**File** menu)

SCREEN NAVIGATION AND EDITING KEYS

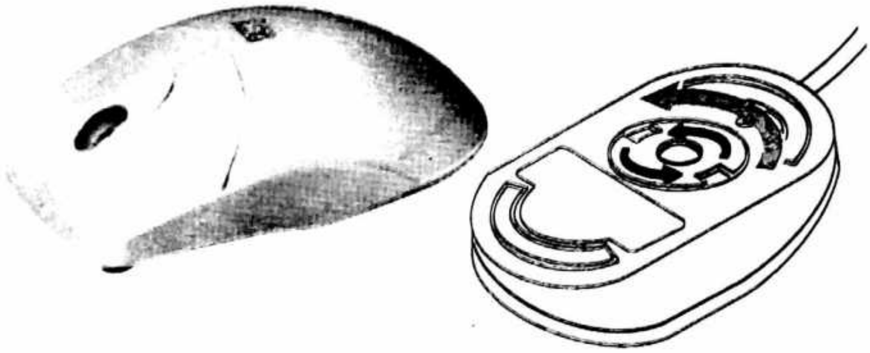
Four cursor control keys are used to move the cursor on the screen in the direction shown by the arrows on these keys. In addition there are special keys for doing special job and editing. For example:-

KEY	FUNCTION OF THE KEY
ESC	Escape key performs a variety of functions defined by the operating system.
CTRL	Perform a command or function when used in conjunction with another key.
SHIFT	Used to change lower case letters to capitals and vice versa. All other keys when pressed with shift key cause the character shown on the upper portion to be displayed.

Backspace	Moves the cursor one character to the left and erases the character in that position whenever the key is pressed.
Enter	Moves cursor from final character position on a line to the first position on the next line.
Caps Lock	When pressed once, locks the characters A to Z in the uppercase position. Press it again for lowercase position.
Num. Lock	Pressing the Number Lock Key once places keys 0 to 9 in Numeric mode. Pressing it again returns it to cursor control.
Delete	The characters at the cursor position is deleted.
Page Up	Page up is a programme controlled key, used to move the cursor to the previous page.
Page Down	Page Down is also a programme controlled key. It is used to move the cursor to the next page.

3.3 THE MOUSE

A mouse is a useful input device. It is a pointing device used to select various options. A mouse is a small hand-held unit with two or three buttons as shown in Figure 3.4(a). It can roll over a small ball at its bottom shown in Figure 3.4(b). When moved over a surface or pad, it moves a small object (often an arrow ↖) across the screen. A mouse can also be used to draw pictures on the screen and edit text.



(a) (b)

Figure 3.4 (a) A typical mouse (b) Bottom view

3.4 JOY STICK

Joystick is an input device, a small box with moving stick and buttons. It is generally found with microcomputers and is used for playing computer games, educational software and computer aided design (CAD) systems. You manipulate the handle of the joystick to position the cursor and click on a button. The joystick does not interfere with your view of the screen nor does it require the movement of an object like a mouse. A joystick is very popular among kids due to the speed with which it moves the objects on the screen.

Figure 3.5 A joystick

3.5 SCANNERS

Image scanner is a useful input device. It converts every bit of the scanned material into electrical pulses. Suitable scanning software converts the printed documents, pictures or photographs into digital files. The digital files so formed can be stored in computer's memory. It can be accessed from computer's memory and imported into a desktop publishing document by suitable graphic software.

Optical character recognition (OCR) software translates the scanned document into text that can be edited. To edit photographs, graphic software are used. You can insert a picture into a text. For this, **scan** the required picture, **copy** it, and then **paste** it in your document. Scanners come in a variety of forms

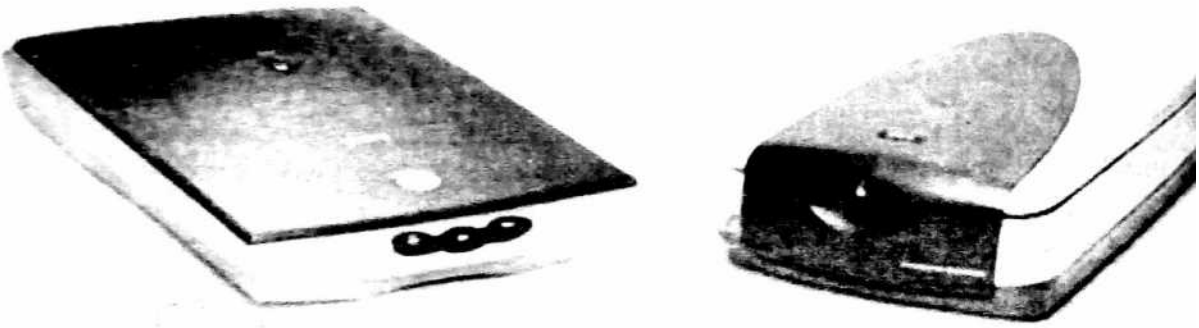


Figure 3.6 Flatbed Scanners

such as full-size flatbed scanners, hand-help scanners, drum scanners, and overhead scanners or film scanners.

3.6 TRACKBALL

A trackball is a pointing device almost like a mouse turned upside down. The user controls the cursor on the screen by rolling a plastic ball with a fingertip or wrist. To execute commands with a trackball, one or more buttons are pressed, much in the same way as is done with a mouse. The cursor can be moved around on the screen by rolling the ball with a thumb or finger.

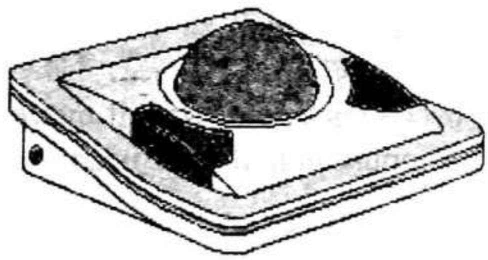


Figure 3.7 A Trackball

Trackball is popular among users of laptop computers when space is limited and may be mounted on either side of the keyboard. For handicapped people who may have difficulty pressing keys on a standard keyboard or using a mouse, the trackball may be the answer since it does not require to move the entire arm to use it.

3.7 ELECTRIC LIGHT PEN

A light pen is an input device attached to some video display terminal. It consists of a photo-cell placed in a small tube which can sense the position on display tube when tip of the pen is held against the screen. In specially designed

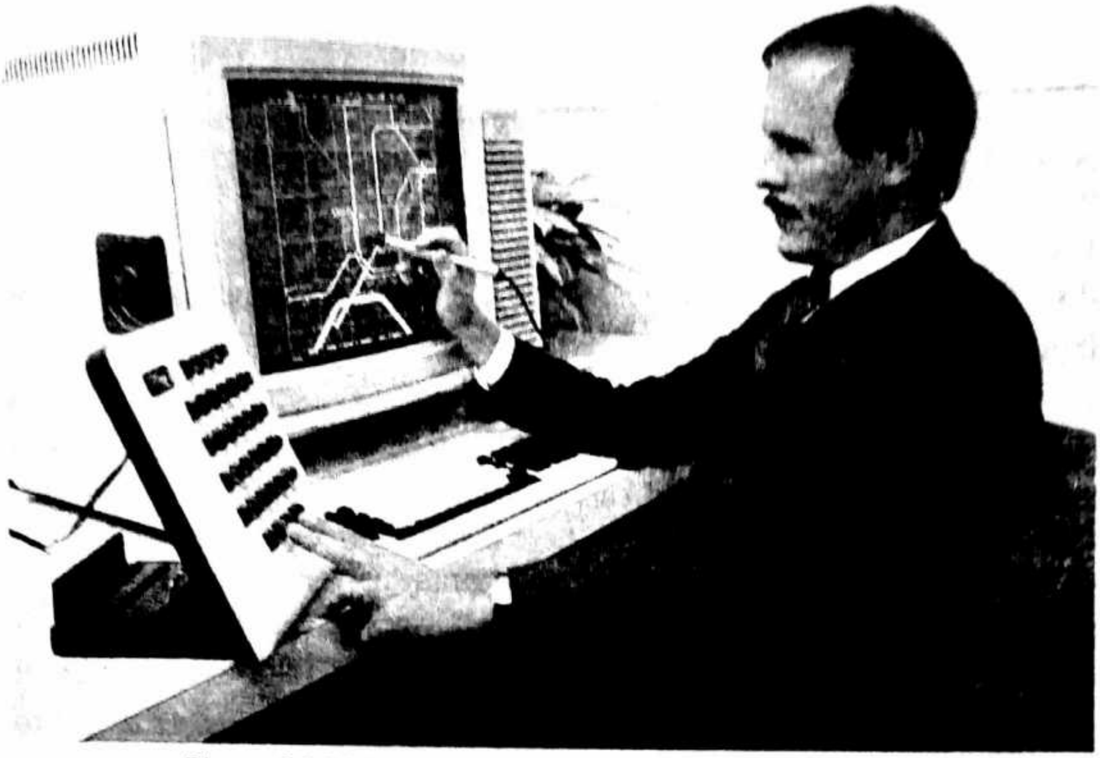


Figure 3.8 Light pen used with graphic display terminals

display screen, data can be entered with a light pen as shown. or existing data can be modified. Figure 3.8 below shows the use of a light pen in conjunction with a graphic display terminal to enter information into the computer.

3.8 MICROPHONES AND VOICE RECOGNITION

Today's computer are capable to work as you speak before it. For this type of sound input, you need a microphone and a sound card. Microphone converts the sound into electrical signals. These electrical signals are translated by the sound card into digitized form that the computer can store and process. Sound cards can also translate digitized sounds back into analog signals that can then be sent to the speakers.

Translating voice to text is a capability known as voice recognition (or speech recognition). With it you can speak to the computer rather than entering the data and command by typing. You can control the computer with simple vocal commands such as **shut down** or **print**.

A speech recognition system contains software. Once you have installed the software, you would need to train the system to recognize your unique speech pattern. To train the system, we simply talk into a microphone attached with it for at least 20 minutes. The system learns our speech patterns and updates the vocabulary accordingly.

3.9 DIGITAL CAMERA

Figure 3.9 A digital Camera

A digital camera converts video images in digital form and saves them on computer's memory. Once on the disk, the digital images can be loaded to a computer and like other graphic images, can be edited also. The images on a computer can be used in a number of ways in various applications.

3.10 DISK DRIVE

The two most popular types of magnetic storage media are the diskettes (commonly known as floppy disks) and the hard disks. Data stored on these disks is often referred to as **softcopy** and can be accessed and altered easily. A **disk drive** is used to **read** and **write** data on these disks. Thus **disk drives** are known as **I/O** devices. A disk drive that rotates the disk may appear to be fairly a simple device. But it is a complex system of several devices that work together. The drive rotates the disk placed in it with precise timing. A **write-head,** which floats

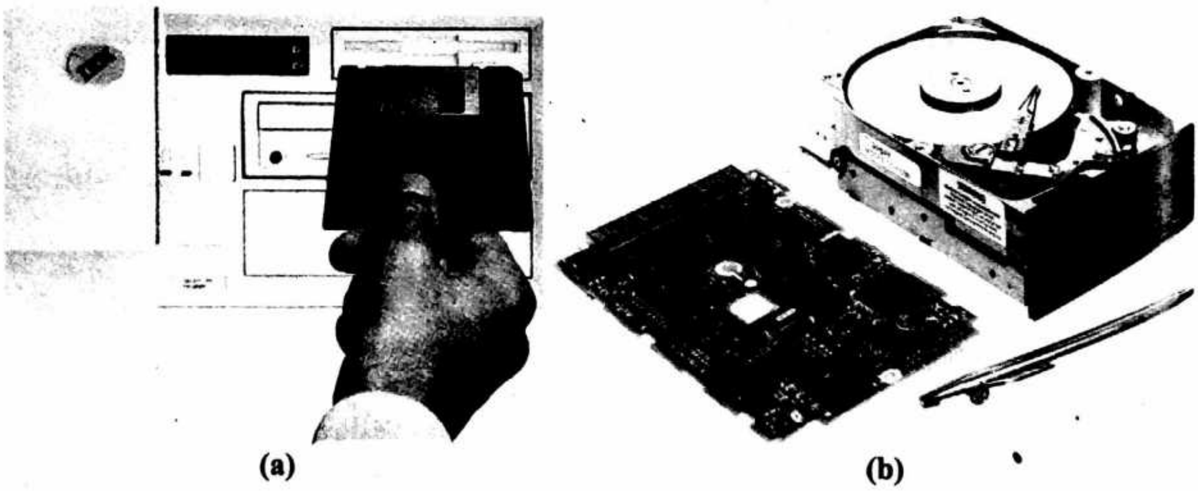

(a) **(b)**

Figure 3.10 (a) Floppy being inserted into its drive **(b)** Hard disks (*right*) under the direction of a controller board (*left*) help to input/output large amounts of data

just above the spinning disk, encode data on the disk. Disk drives are used to input extremely large amount of data and instructions in a few seconds. Copies of the coded data and instructions can also be prepared easily on other disks for use in other systems.

3.11 OUTPUT DEVICES

Output devices provide ways to get back the information from a computer. A number of output devices are in use such as monitors, printers, disk drives etc.

3.12 MONITORS

Monitor also called as video display unit is the most popular output device. A monitor displays information on the screen of a picture tube similar to the one used in a television set. Picture or text that appears on the screen is formed by tiny dots called picture elements or **pixels**. These **pixels** determine the quality or resolution of the image. Greater is the number of pixels higher will be the resolution and better will be the image quality.

Figure 3.11 A coloured monitor

There are two types of monitors, black and white and coloured monitors, both are available in different sizes. Figure.3.11 shows a colour monitor. Colour monitors can display images in multiple colours. Monitors are classified by their image producing technology. For example a **colour graphic adapter (CGA)** displays four colours at a resolution of 320 by 200 pixels. An **extended graphic adapter (EGA)** produces images upto 16 different colours at a high resolution of 640 by 480 pixels. A **video graphic array (VGA)** presents simultaneously up to 256 colour shades at resolutions up to 720 by 400 pixels. Similarly new monitors of high resolutions are coming up with new standards such as **S-VGA (super VGA)** from 800 by 600 pixels to 1280 by 1024 pixels.

Now-a-days LCD flat-panel coloured monitors are becoming popular. LCD coloured monitors are particularly suitable with laptop PC because they are light weight, portable and use little power.

3.13 PRINTERS

The printers are one of the most useful output devices for use with a computer. In the early days of computing, it was the only source of information about what the computer had done. Printers are used to produce **hard copy** or permanent record of computer's output. These are classified into two main categories depending upon the type of print mechanism used in a printer.

- Impact Printers
- Non-Impact Printers

3.14 IMPACT PRINTERS

Impact printers work like typewriters. Impact printers come in three types based on their speed of printing and number of characters per line. These are:-

- Dot Matrix Printers
- Daisy Wheel Printers
- Line Printers

DOT MATRIX PRINTERS

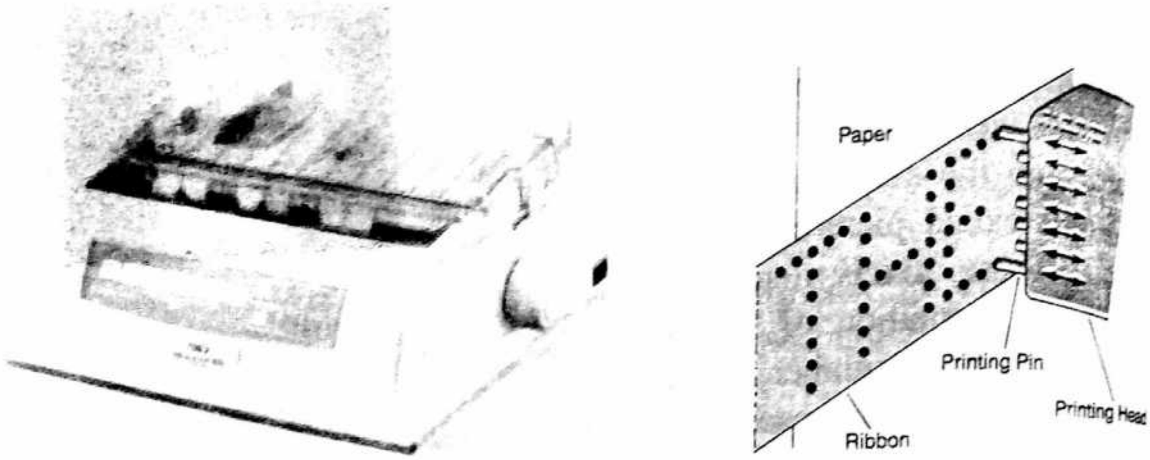


Figure 3.12 (a) A dot matrix printer. **(b)** Patterns formed by Dot matrix head.

The dot matrix printer is an impact printer as shown in Figure.3.12.(a) I uses tiny pins arranged to hit the ribbon and the paper. In dot matrix printer, each letter is formed with the combination of dots as shown in Figure 3.12.(b) A do matrix may have 9, 18, 24 or 32 pins arranged vertically in a print head. The do matrix printer arranges pins to print dots that form characters and all kinds o images. A normal dot matrix printer has speeds varying from 50 to 600 character per second. The carriage width of a printer also varies ranging from 80 columns t 132 columns. A standard-width printer prints 80 characters per line.

DAISY WHEEL PRINTERS

The daisy wheel printer uses a flat disk or wheel with petal like projections each with a different character at the tip as shown in figure 3.13. The whee rotates to bring the desired character into position. A character is formed whe hammer strikes the petal under it. Daisy wheels are available in several formats (bold letter, italics etc.). A daisy wheel printer is slower than a dot matrix machine As print wheel takes more rime to bring the desired petal in position. However it produces high-quality print.

Figure 3.13 Daisy Wheels and Daisy Wheel Print Mechanism

LINE PRINTERS

A number of different types of high speed printers have been developed to satisfy the needs of large computer systems. These are called line Printers as they print one complete line at a time. Line printers are also impact printers and may be further divided into drum printers and band or chain printers. Some of these printers can produce over 3000 lines per minute.

3.15 NON-IMPACT PRINTERS

Non-impact printers have been developed which produce a printed image without striking the paper. These printers use new methods that have improved printing quality and speed. Non-impact printers are noiseless and faster than impact printers. Some non-impact printers are: -

- **Electrostatic printers.**
- **Ink jet printers.**
- **Electro thermal printers.**
- **Electrostatic printers.**

ELECTROSTATIC PRINTERS

Electrostatic printers are high-speed non-impact line printers. A charged image is formed on paper by electric field. This paper is passed through an ink-fog. The ink particles adhere at the image on the paper to form a visible image. The paper is then passed through the heated rollers for fixing the image.

ELECTRO-THERMAL PRINTERS

Thermal printers produce characters on heat sensitive waxy paper. These were popular a few years ago because they are portable and low cost. But the drawback is that they require special heat sensitive waxy paper.

INK JET PRINTERS

Ink jet printers work in the same way as dot matrix printers except that ink jet printers have fine nozzles instead of tiny pins used in dot matrix print head.

Nozzles spray a stream of ink onto the paper. Because the ink is put directly on the paper, therefore, these printers require ink in reservoirs instead of ribbon. Since no mechanical movement is involved, hence these printers are much quieter and have speed of about 200 characters per second. Ink jet printers are also available which have more than one ink reservoir, each with a different colour. The colours can be chosen through software to allow the printer to print in rainbow colours. Thus a full colour graphic image fro computer screen can be obtained very quietly on paper.

Figure 3.14. A coloured Ink Jet Pri with a colour print coming o

LASER PRINTERS

Laser printers are the fastest non-impact printers. These printers use laser and electro-photographic technology and have printing speeds greater than 20,000 lines per minute. A bit-mapped image of a page is formed in the computer by software. This image is transferred to printer which activates a laser beam. The laser beam duplicates the image o rotating drum and forms a charged image on it. The image is transferred on paper that passes between the drum and a positively charged corona wire. T dots of the toner forming the image get fused onto the paper under the combir effect of heat and pressure when passing through the roller. The printed res from a laser printer is impressive and better in quality.

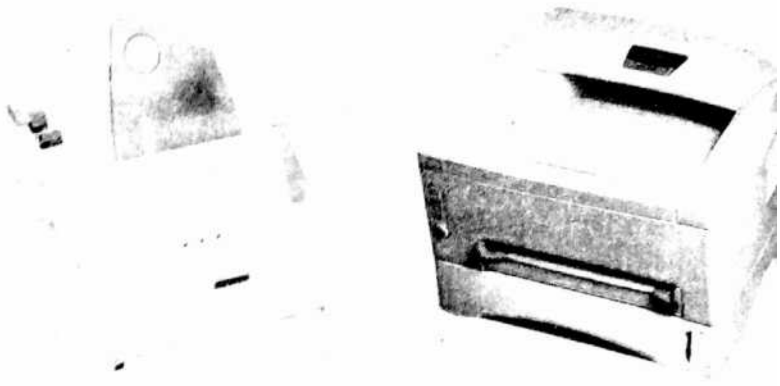


Figure 3.15 Laser Printers

3.16 PLOTTERS

Plotters are special output devices that print out black/white or colour graphics output. Plotters produce large drawings or images such as constructi plans of buildings or blue prints of complex machines like aircraft design.

The two basic types of plotters are the **drum plotter** and the **flatbed plotter**. Both have one or more pens that move over a sheet of paper to produce images. In a drum plotter, the pen and the drum move along **x** and **y-axes** to trace out the pattern, graph or chart. Drum plotters are also used to produce continuous output. A flatbed or **x-y** plotter uses drawing arm to draw image on paper rolled out onto a flat surface. The pen moving along **x** and **y-axes** can mark any position on the paper. Multicoloured graphs and charts can be developed using several pens of coloured ink.

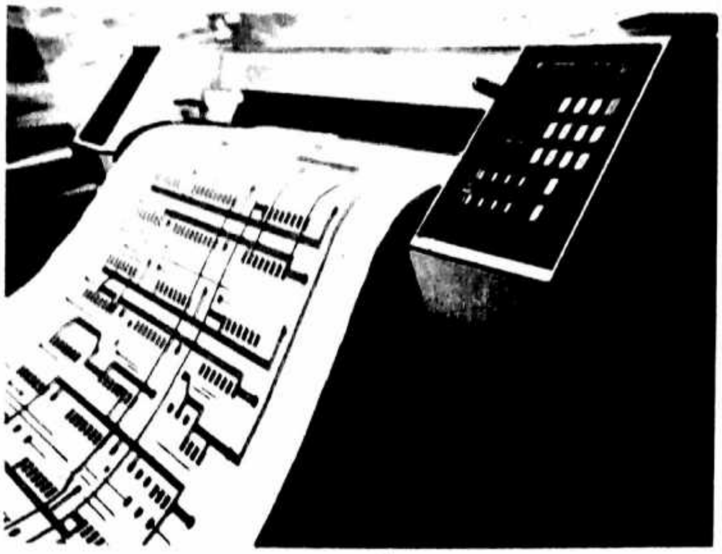


Figure 3.16 A Plotter with a coloured graphic output

3.17 CD WRITER

A CD-writer is an output device. It can write once on a **recordable compact disk** (CD-R) to create a CD-ROM. The rapid and universal acceptance of **compact disk read only memory** (CD-ROM) has given rise to an exciting new technology. CD-ROM is the backbone of multimedia applications which involve the combination of text, sound, graphics, motion video and animation. Nearly all the PCs have CD-ROM drives. A locally developed CD-R (compact disk recordable) can play in any CD-ROM drive. A CD-R is written by CD writers. CD writers offer a low-cost alternative in making **softcopies** by saving large amount of data on CDs. The data cannot be changed once written on a disk. It can only be read from the disk.

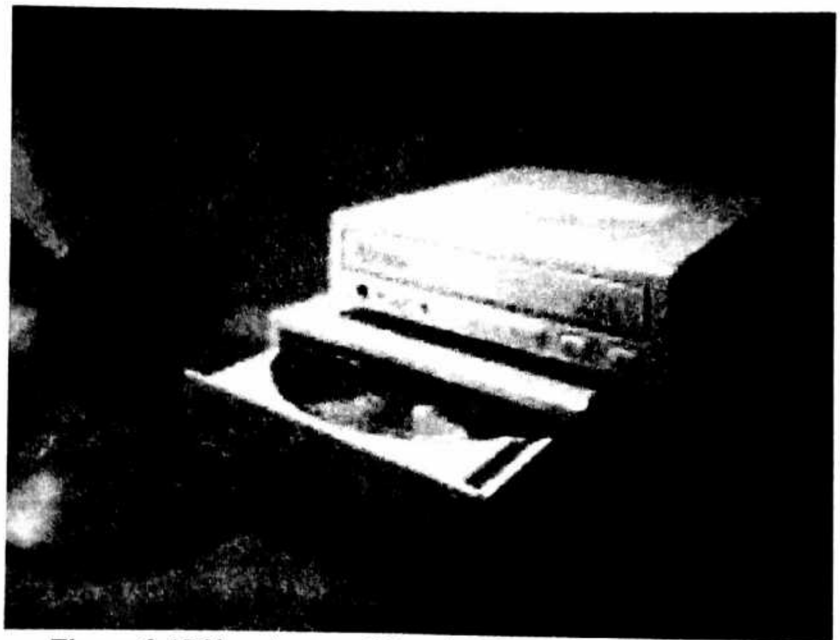


Figure 3.17 Showing two CD writers with CD-ROMs in the background. A CD-R is just inserted in the lower CD writer

3.18 DISK DRIVE

It has already been discussed that disk drives are **input/output devices**. A disk drive which can read data from a floppy disk or hard disk can also write data on these disks. A drive rotates a disk with precise timing. A **read/write-head** floats above the surface of the spinning disk having magnetically encoded data. **Read-head** picks up large amount of data stored on the disk. The data read by the read-head goes into the computer's main memory Disk drives are used to input extremely large amount of data and instructions in a few seconds. Copies of the coded data and instructions can also be prepared easily on other disks for use in other systems.

3.19 HARD-COPY/ SOFT-COPY

In computer, the term **hard-copy** is used for Computer's output that has been printed on some surface like paper. Once its printout is obtained, the printout cannot be changed and is a permanent record in usable form. Thus printers produce **hard copy** or **printed output** On the other hand computer's output that appears on a monitor or on a screen as a video image or is saved on some medium such as disks is called a soft-copy. In other words any other form that is not hard copy is called **soft copy.** The material printed on paper sheets**, like book, newspaper etc. is the hard-copy.** The machine's output displayed on a screen or the files saved on floppy or hard disk is the **soft-copy.**

SUMMARY

Input devices send data and instructions from outside to CPU. The CPU processes the data as directed. **Output devices** retrieve processed information from the computer's memory and display it on the screen, print it out or save it for future use. Without Input/Output devices, we cannot communicate with computer.

INPUT DEVICES

Input devices are used to feed data into the computer and also to instruct it what to do? Some of the common input devices are keyboard, mouse, joystick, optical character reader, light pen, scanner, disk drive etc.

KEYBOARD

The keyboard is an important input device used to enter data and instructions into a computer. A standard arrangement of keys is the QWERTY arrangement. Popular keyboards have enhanced QWERTY arrangement.

The two basic styles of keyboard are PC/XT-style keyboards and AT-style keyboards. A keyboard may be divided into four general areas; Alphanumeric keypad, Numeric keypad, Function keypad, Screen Navigation and editing keys.

THE MOUSE

A mouse is a pointing input device used to select various options. It is a small hand-held unit with one, two or three buttons. It can roll over a small ball at its bottom. It can also be used to draw pictures on the screen and edit text.

JOY STICK

Joystick is an input device, a small box with moving stick and buttons. It is generally found with microcomputers for playing computer games

SCANNERS

Image scanner is a useful input device. Suitable scanning software converts the printed documents, pictures or photographs into digital files. Optical character recognition (OCR) software translates the scanned document into text that can be edited. To edit photographs, graphic software is used.

TRACKBALL

A trackball is a pointing device almost like a mouse turned upside down. The cursor moves on the screen by rolling the plastic ball with a fingertip or wrist.

ELECTRIC LIGHT PEN

A light pen is an input device attached to some video display terminal. It consists of a photo-cell placed in a small tube which can sense the position on display tube when tip of the pen is held against the screen.

MICROPHONES AND VOICE RECOGNITION

Today's computer is capable to work as you speak before it. For this type of sound input, you need a microphone and a sound card. Translating voice to text is a capability known as voice recognition. With it you can speak to the computer rather than entering the data and command by typing. A speech recognition system learns our speech patterns and updates the vocabulary accordingly.

DIGITAL CAMERA

A digital camera converts images in digital form and saves them on computer's memory.

DISK DRIVE

The two popular types of magnetic storage are the diskettes and the hard disks. Disk drives are known as I/O devices. The drive rotates the disk put into it with precise timing. A read/write-head, which floats just above the spinning disk, reads/writes data on the disk. Disk drives input large data in a few seconds.

OUTPUT DEVICES

Output devices provide ways to get back the information from a compu A number of output devices are in use such as monitors, printers, disk drives et

MONITORS

Monitor also called as video display unit is the most popular output dev Number of **pixels** on the screen determine the quality or resolution of the ima **S-VGA** monitors are of high resolutions from 800 by 600 to 1280 by 1024 pixe

PRINTERS

The printers are one of the most useful output devices. Printers are used produce **hard copy** or permanent record of computer's output. These classified into two main categories; Impact Printers and Non-Impact Printers

IMPACT PRINTERS

Impact printers work like typewriters. Various types of Impact printe are; Dot Matrix Printers, Daisy Wheel Printers and Line Printers.

NON-IMPACT PRINTERS

Non-impact printers produce a printed image without striking the pape They have improved printing quality and speed. Some non-impact printers ar Electro thermal printers, Ink jet printers and Laser printers.

PLOTTERS

Plotters are special output devices that produce graphics output in blac and white or in colour. Plotters produce large drawings or images such construction plans of buildings or aircraft design etc.

CD WRITER

A CD-writer can write once on a **recordable compact disk** (CD-R) create a CD-ROM. A CD-ROM can save large amount of data upto 700 Mbytes.

DISK DRIVE

A drive rotates a disk with precise timing. A **read/write-head** floats abo the surface of the spinning disk and picks up data stored on the disk.

HARD-COPY/ SOFT-COPY

In computer, the term **hard-copy** is used for Computer's output that ha been printed permanently on some surface like paper, fabric sheet, film etc. O the other hand computer's output that appears on a monitor or on a screen or saved on some medium such as disks is called a soft-copy. The contents of a so copy can be altered easily at any time. In other words any other form that is not hard copy is called a **soft copy.**

EXERCISES

3.01 Complete the following statements.

i) A standard arrangement of keys on a keyboard is the__________ .

ii) A mouse is a pointing device used to select__________________.

iii) A trackball is almost like a ______________ turned upside down.

iv) Translating voice to text is a capability known as _____________.

v) VDU is an ____________ device.

3.02 Tick (✓) the following statements either True or False.

i) Input devices are used to enter only data into a computer. True/False

ii) Alphabet keypad consists of keys from A to Z and 0 to 9. True/False

iii) A scanner picks every bit of the scanned text. True/False

iv) The disk drives are known as I/O devices. True/False

v) A CD-Writer is an I/O device. True/False

3.03 Encircle one choice A, B, C or D in each case.

i) Which of the following is an output device.

(A) Keyboard (B) mouse (C) disk (D) scanner

ii) Spacebar on a keyboard is a part of the:-

(A) Alpha numeric keypad (B) Numeric keypad

(C) Function keys (D) Screen Navigation & editing keys

iii) Which of the following is not a pointing device.

(A) Mouse (B) Joystick (C) Trackball (D) non of these

iv) Which of the following is used to produce soft copy.

(A) Laser Printer (B) Plotter

(C) CD-writer (D) Dot-Matrix printer

v) The number of colours a CGA monitor displays are:-

(A) 4 (B) 16 (C) 64 (D) 256

3.04 Match the items given in Column I with those given in Column II

	Column I		Column II
i)	OCR software	a)	Printer
ii)	Monitor	b)	Input
iii)	Disk drives	c)	LCD
iv)	Laser	d)	I/O
v)	Digital Camera	e)	Scanner text

3.05 Name two very important input devices and describe their purpose computer system.

3.06 Name different areas of a keyboard and some key functions.

3.07 What is a mouse? How it works?

3.08 Describe the working and uses of a trackball.

3.09 What do you know about scanner? How scanner text can be edited?

3.10 What is meant by voice recognition? Explain.

3.11 What is a colour monitors? Describe some features of different coloure monitors.

3.12 Explain the difference between Impact printers and Non-Impact printers.

3.13 Explain the working of a dot-matrix printer.

3.14 Explain how Ink-jet printer differs from Laser printer?

3.15 What is a plotter? How it works?

4

STORAGE DEVICES

As you have already learnt in chapter 2 that data and instructions are first placed in Computer **storage** often called **memory** on entering into a computer. They remain there until called for processing by the control unit. Memory or storage is a major factor in computer power. More powerful computers store more data and process large amount of data.

4.1 COMPUTER MEMORY/STORAGE

Computer memory or storage has two major divisions. These are:-

- **Main memory or Internal memory**
- **Secondary memory or Backing storage**.

The main reason for the distinction between main memory and secondary memory is the cost in relation to their performance in handling data and storage capacity. CPU has quick access to main memory. Main memory can store and supply data and instructions at a very high speed. But it is very costly. Secondary memory is much slower and cannot communicate data and instructions at speeds at which CPU processes data. However secondary memory is much cheaper than main memory and provides very high storage capacity.

4.2 MEMORY UNITS

The capacity of computer storage is expressed in number of **bits, bytes** or **characters**. Generally a combination of eight bits represents a character.

BITS AND BYTES

The basic memory unit of a digital computer is a **bit**, which stands for **binary digit**. Bit represents a binary number that is either **0** or **1**. The memory of a computer is composed of **cells**. Each cell contains one bit information. Each memory cell has a value 0 or 1 represented by **ON** or **OFF** state of the cell.

As bit is a very small unit of data, therefore bits are grouped into bigger units. A group of four bits is called a **nibble** and that of eight bits is called a **byte**. A **byte** is generally used to express the memory of a computer.

1 nibble		$= 2^2$ bits	= 4 bits
1 byte		$= 2^3$ bits	= 8 bits
1 kilobytes	(Kbytes)	$= 2^{10}$ bytes	= 1024 bytes
1 megabytes	(Mbytes)	$= 2^{20}$ bytes	= 1024 kilobytes
1 gigabytes	(Gbytes)	$= 2^{30}$ bytes	= 1024 megabytes
and 1 terabytes	(Tbytes)	$= 2^{40}$ bytes	= 1024 gigabytes

THE WORD

Data/instructions flow in groups of bits in computers called the **words.** A **Word size** is the number of bits a microprocessor can manipulate at one time. For example an 8-bit microprocessor can manipulate 8 bits of data at a time. A 16-bit microprocessor can handle 16 bits of data at a time. Thus a 16 bit microprocessor can manipulate twice as much data as an 8-bit microprocessor in one cycle. Computers that handle larger word-size are considered to be faster. Thus a 64-bit microprocessor is faster than a 32-bit microprocessor.

4.3 MAIN MEMORY

Main memory is the computer's internal memory. It is an extension of the Central Processing Unit (CPU). Its function is controlled by the Control Unit (CU) that sends READ or WRITE signal for the appropriate memory location. Main memory accepts data and instructions from the input unit, exchanges them with other parts of the CPU for processing. It stores those instructions and data, which are currently being used by the CPU or awaiting for immediate execution.

Semiconductor memories are the widely used forms of main storage. The capacity of a storage device is normally expressed in **bytes, characters** or **bits**. Computer's main memory is further divided into **random access memory** (RAM) and **read only memory** (ROM)

RANDOM ACCESS MEMORY (RAM)

RAM is that part of main memory in which data and instructions are held temporarily. RAM provides a working area to the user to enter and process data. In RAM each data element has its own address (location). Any data element can be read easily and quickly by using that address. It is also called as Read-And-Write Memory (RAM) since the computer can store or **write** data at any selected location (address) and can retrieve or **read** data when needed. It is a **temporary memory** of a computer used to store data and instructions when the computer is on. Everything that is stored in RAM is lost when the computer is turned off. For this reason it is also called **volatile memory**. Large RAM sizes provide larger data that a computer can hold and process. Additional RAM chips can be installed in a computer simply by plugging them on motherboard. This increases the storage capacity of RAM of a computer.

RAM is a semiconductor memory with no moving part. Data can be accessed from RAM at very high speed very close to the speed of light. A memory chip less than one-fourth the size of a postage stamp can store data more than 500,000 bytes equivalent to the printed matter on a popular daily newspaper!

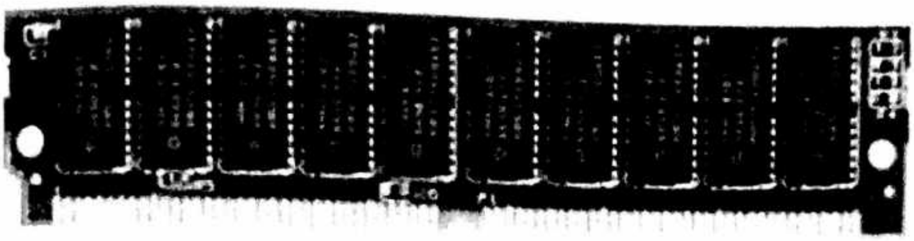


Figure 4.1 Single inline memory module containing ten RAM chips

READ ONLY MEMORY (ROM)

Another important part of main memory is **Read Only Memory** (ROM). Instructions can only be read from ROM. Instructions once written on ROM chip cannot be changed. It is a permanent or non-volatile memory of a computer, i.e. the contents of this type of memory are not lost when the computer is turned off. As soon as the computer is **turned on,** a program that resides in ROM automatically makes the computer ready for use. ROM is a semiconductor memory generally programmed by the manufacturer.

SINGLE IN-LINE MEMORY MODULES (SIMMs)

The capacity of RAM in a computer affects computer's power. More RAM can make computer run faster. The computer does not necessarily have to load a complete program into its main memory to run it. However greater is the part of a program it can fit into memory, the faster the program will run.

Many RAM chips are installed on a small circuit board. These RAM chips are wired together to form a single module of large memory called **Single In-line Memory Modules** (SIMMs). SIMM has contact terminals on its one side. It can easily be plugged into the motherboard. SIMMs had different capacities such as 1 Mbyte, 4 Mbytes 16 Mbytes or 32 Mbytes with varying speed.

DUAL IN-LINE MEMORY MODULES (DIMMs)

SIMMs were popular by PC users in 1980s but has now been replaced by **DIMMs**. Like SIMMs, **Dual In**-line **Memory Modules** (DIMMs) have been developed installing many RAM chips. DIMMs provide larger capacity as compared to SIMMs. DIMMs have wider data bus and thus have higher rate of data transfer. DIMMs are available in several configurations having different capacities and speeds. You can expand RAM capacity of your computer by simply plugging-in more DIMMs.

4.4 SECONDARY MEMORY

Secondary memory is also called the backing storage because it is used supplement the capacity of main memory. Secondary memory is essential only due to the limited size of RAM but also because it is a temporary memory.

Secondary memory provides large capacity for storing large number programs and hundreds of Mbytes of data. Examples of secondary memories magnetic tapes, magnetic disks and mass cartridge systems. Secondary stora are either **sequential access** type or **direct access** type.

SEQUENTIAL ACCESS STORAGE

In sequential access all the data recorded on the storage is examin sequentially. An operator cannot refer directly to the contents of a particu storage location. It starts reading record from the beginning until desired data found. Magnetic tapes, Audio and Video cassettes are sequential access storages

DIRECT ACCESS STORAGE

Direct access devices provide immediate access to individual records a do not require reading from the beginning of a file to find a particular recor Direct Access Storage is addressable. A given item can be accessed fro anywhere in the storage by simply addressing its location. Examples of dire access storages are; Magnetic disks, Compact disks etc.

4.5 MAGNETIC TAPE

Magnetic tapes varying in width from ½ inch to 1 inch wide of differe lengths held on a plastic spool are used. Audio cassette tapes 3.175 mm to 6.3 mm (1/8 inch to 1/4 inch) wide are also used to store programs. These are chea and reliable. But they take considerable time to load and to locate a programme.

4.6 MAGNETIC DISKS

A **magnetic disk** is a metal or plastic disk coated with magnetic materia Data is recorded onto the magnetic material in machine code. Disks have becom popular due to their random access. These disks spin in the **disk drive** **Read/write** heads moving on the disk along its radius can read data at an location under the heads.

Different types of disk drives and magnetic disks are in use. The mos popular types of magnetic disks are interchangeable disks called the **floppy disk** and fixed disks called the **hard disks**. Generally PCs are configured with at leas one hard disk drive and one floppy disk drive. Floppies are important fo transporting data and software. The high-capacity hard disks in PC have made i possible to keep all the data and software readily accessible at all times.

FLOPPY DISKS

Floppy disks are the most common and popular type of secondary storage. Floppy disks are flexible made of oxide-coated Mylar. These are stored in a paper or plastic envelopes. The data on it can be accessed randomly. You can copy, change, add, delete or rename a file. Thus anything you can do with a paper file can also be done with a disk file

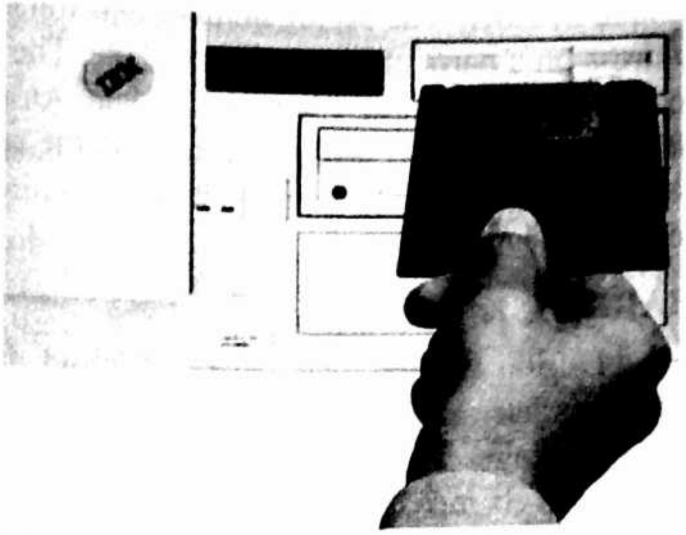


Figure 4.2 The popular 3½ inches floppy disk and the drive.

Floppy disks have three different sizes 8", 5.25" and 3.5" diameters. floppy disks of 8" and 5.25" have now been outclassed. A micro-floppy of 3.5" diameter shown in Figure 4.2 can hold upto 1.44 Mbytes. Generally micro-floppy drive is assigned **drive A** and mini-floppy drive is assigned **drive B**.

The floppy cover has a window with a spring-loaded metal shutter. The shutter is pushed back which uncovers the window when it is inserted into the disk drive. The drive rotates the disk inside its protective covering at a speed of 300 rpm. Read/write head contacts its exposed surface through the **window**. Recording is done magnetically in concentric circles called tracks as shown in Figure 4.3 Data is read or written serially in bits on the tracks within a given sector.

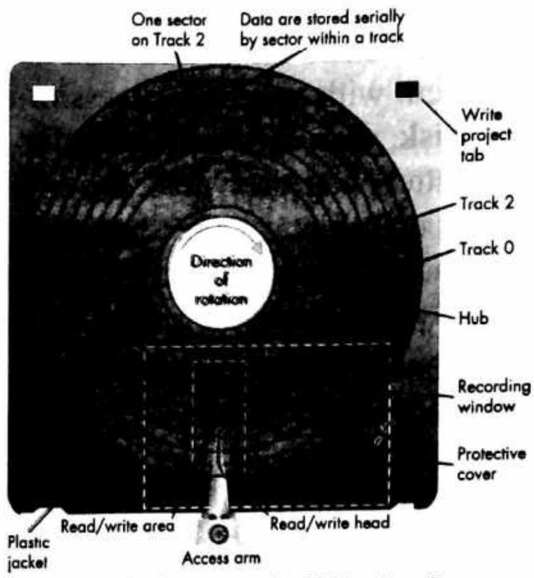


Figure 4.3 Cutaway of a 3½ inches floppy

Microcomputer disks use sector organization to store and retrieve data. In sector organization, the recording surface is divided into pie-shaped sectors. The number of sectors depends on the density of the disk. Each sector is assigned a

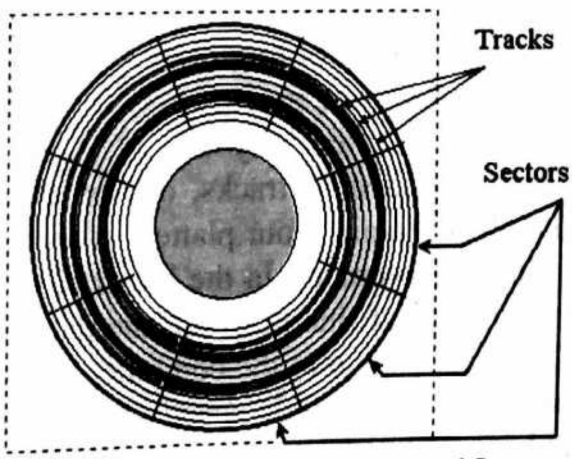


Figure 4.4 Showing concentric Tracks and Sector on a Floppy Diskette.

unique number. The sector number and track number are all that are needed for address on a particular disk-face surface. The disk address represents the physical location of a particular file or set of data. An access arm containing the read/write head is moved under program control to the appropriate track see Figure 4.3. Data are read or written when a sector containing the desired data passes under the read/write head.

HARD DISK

During the last few years. the rapid development in application software packages has made the users' work much easy and simple. But at the same time these packages has increased the need of large storages. **Hard disks** have been developed to meet the growing demand for secondary storages. These are high speed, large capacity disks and are referred as **mass-storage** magnetic medium. Hard disks available now-a-days have memory upto tens of Gbytes.

Hard disks are coated with magnetic material on their surfaces and provide the system with ability to access or save information sequentially or randomly. A **hard disk**, also called **Winchester disk**, consists of one or more rigid metallic disk platters and their associated **read/write** heads encased in a sealed chamber. All the disk platters are stacked on a common rotating spindle. Data and the instructions are recorded on both the surfaces of a platter.

For a hard disk with four platters, there are eight recording surfaces on which data can be stored with eight read/write heads as shown in figure 4.4. As all the heads move together so all are at the same track number on their respective recording surfaces at the same time. A set of similarly numbered concentric tracks, one on each surface forms a **cylinder**. Thus, for a hard disk containing four platters, a computer can access a cylinder of eight tracks in a single movement. In the illustration, the read/write heads are positioned over cylinder 0012 at this position, the data on any one of the eight tracks numbered 0012 are accessible to the computer on each revolution of the disk. The disks spin continuously at a high speed (from 3600 rpm to 10000 rpm within a sealed chamber). The chamber keeps the disk surfaces free from dust and smoke.

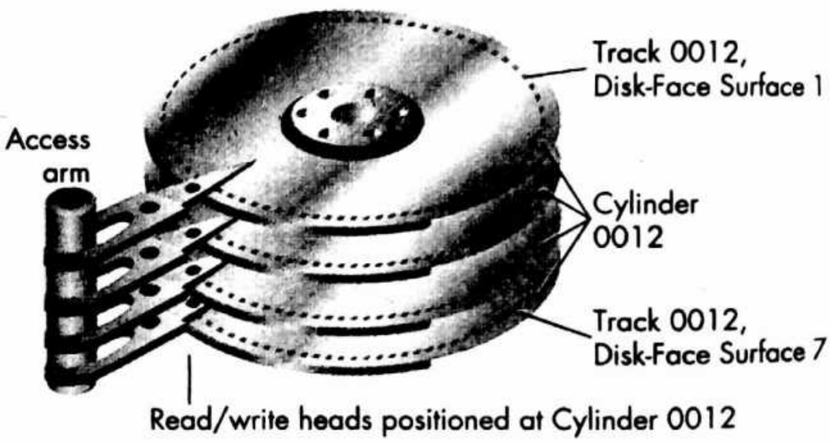


Figure 4.4 Fixed Hard Disk with Four Platters and Eight Recording Surfaces

4.7 COMPACT DISK (CD-ROM)

CD-ROM stands **for Compact disc-read only memory**. These discs are becoming popular now-a-days especially for use with microcomputers. The name implies to its application. Once inserted into the CD-ROM drive, the text, video images and so on can be read into RAM for processing or display. However the data on these disks are fixed and cannot be altered in contrast to read/write capability of magnetic discs. CD-ROMs have very large storage capacity. The capacity of a single CD-ROM is upto 680 Mbytes equivalent to 500 floppies.

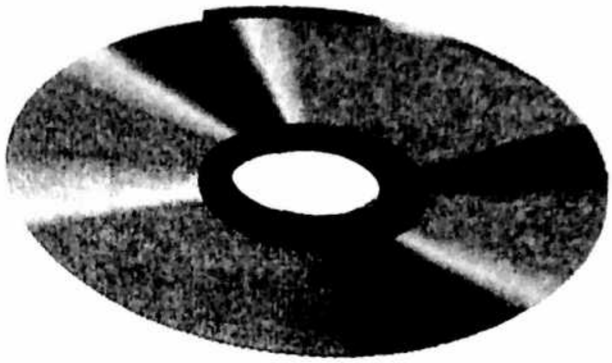


Figure 4.5 A Compact Disk Read Only Memory.

Unlike magnetic disks that store data in concentric tracks, a CD-ROM stores data on a single track that spirals from the centre to the outside edge. This ultra thin track spirals around the disk thousands of times.

Data are recorded on CD ROMs' reflective surfaces in the form of pits and lands. The pits or depressions are tiny reflective bumps that have been burned in by a laser beam. Lands are flat areas separating the pits. A land reflects the laser light into the sensor and a pit scatters the light. A spot that reflects the laser beam into the sensor is interpreted as **1**. A spot that scatters and does not reflect the laser beam into the sensor is interpreted as **0**.

The fact that you cannot write data on a CD-ROM does not mean that this storage medium is not useful. In fact many applications rely on huge volumes of data that rarely require changes. For example, dictionaries, encyclopaedias, medical, legal and other professional reference libraries, software packages, music and videos all require huge memory that you normally would not want to alter.

SUMMARY

Storage capacity of a computer is an important factor. More powerful computers can store and process large amount of data.

COMPUTER MEMORY/STORAGE

Computer memory or storage has two major divisions; Main memory or Internal memory, Secondary memory or Backing storage. The capacity of computer storage is expressed in number of **bits, bytes** or **characters**.

BITS AND BYTES

The smallest memory unit of a digital computer is a **bit**. A group of fo bits is called a **nibble** and a group of eight bits is called a **byte**. A **byte** is T memory of a computer is generally expressed in bytes and multiples of bytes.

THE WORD

In computers, data/instructions flow in groups of bits called **words**. **Word size** is the number of bits a microprocessor handles at one time. Comput that handle larger word-size are considered to be more powerful. Since larg amount of data can be moved and processed in each cycle.

MAIN MEMORY

Main memory is the computer's internal memory. Main memory accep data and instructions from the input unit, exchanges them with other parts of t CPU for processing. Main memory is further divided into RAM and ROM.

RANDOM ACCESS MEMORY (RAM)

RAM is that part of main memory where data and instructions are he temporarily during execution. It is a **temporary memory** and is also called **volatile memory.** RAM is a high speed semiconductor memory.

READ ONLY MEMORY (ROM)

Another important part of main memory is ROM. Once written on a ROI cannot be altered. It is a permanent memory supplied by the manufacturer.

SINGLE IN-LINE MEMORY MODULES (SIMMs)

Many RAM chips installed on a small circuit board are wired together t form a SIMMs. More RAM makes computer to run faster.

DUAL IN-LINE MEMORY MODULES (DIMMs)

SIMMs has now been replaced by DIMMs as they have large storage capacity and higher rate of data transfer than SIMMs.

SECONDARY MEMORY

Secondary memory is used to supplement the capacity of main memory. I provides large storage capacity. Secondary storages are either **sequential acces** type such as Magnetic tapes, Audio and Video cassettes. or **direct access** type such as Magnetic disks. Compact disks and Laser holographic storage

MAGNETIC DISKS

A **magnetic disk** is a metal or plastic disk coated with magnetic material. These disks spin in the **disk drives. Read/write** heads moving on the disk can read/write data at any location under it. The most popular types of magnetic disks are interchangeable disks called **floppy disks** and fixed disks called **hard disks.** Microcomputer disks use concentric tracks and sector organization to store and retrieve data.

COMPACT DISK (CD-ROM)

CD-ROM are becoming popular now-a-days. A CD-ROM stores nearly 700 Mbytes data on a single track that spirals from the centre to the outside edge. This ultra thin track spirals around the disk thousands of times. Data are recorded on CD ROMs' reflective surfaces in the form of pits and lands. The pits or depressions are tiny reflective bumps that have been burned in by a laser beam.

EXERCISES

4.01 Complete the following statements

i) A byte is a group of ________ bits.

ii) ________ is a volatile memory.

iii) Storage capacity of a sector on floppy is a multiple of ______ bytes.

iv) SIMMs provide________memory capacity as compared to DIMMs.

v) The capacity of a CD of 700MB is equivalent to ________ floppies.

4.02 Tick ( ) the following statements either True or False.

i) ROM is a part of computer's internal memory. True/False

ii) A DIMM contains many ROM chips. True/False

iii) Hard disk is a sequential access storage. True/False

iv) Number of tracks on a CD is greater than a floppy. True/False

v) A pit on a CD reflects laser light while a land scatters. True/False

4.03 Encircle one choice A, B, C or D in each case.

i) As compared to the main memory, secondary memory of a computer

(A) has faster access (B) has smaller capacity

(C) is cheaper (D) resides in CPU

ii) The data from RAM can be accessed at a speed close to

(A) Supersonic speed (B) Speed of light

(C) Speed of sound (D) Speed of ultrasonic

iii) The number of read/write heads for a hard disk of four platters is

(A) 4 (B) 6 (C) 8 (D) 16

iv) The number of tracks in one cylinder of hard disk with two platters is

(A) 4 (B) 8 (C) 16 (D) 32

v) A hard disk is also called a

(A) Compact disk (B) Winchester disk

(C) System disk (D) Changeable disk

4.04 Match the items given in Column I with those given in Column II

Column I		Column II	
i)	Nibble	a)	Speed
ii)	Word	b)	Laser
iii)	16 MB RAM	c)	Memory unit
iv)	128 MB RAM	d)	SIMMs
v)	CD-ROM	e)	DIMMs

4.05 What is meant by computer storage? How will you classify it?

4.06 What is smallest unit of memory in digital computers?

4.07 Name 4 memory units in which memory of a storage device is measured?

4.08 What is the significance of byte? How other memory units are related with byte?

4.09 Name some of the computer's primary and secondary storage devices.

4.12 What do you know about RAM?

4.13 In what ways RAM and ROM differ?

4.14 What is ROM? How do PROM and EPROM differ from each other?

4.15 Differentiate between SIMMs and DIMMs.

4.16 Name the types in which magnetic disks can be divided.

4.17 How data can be written on or retrieved from a floppy?

4.18 Why the capacity of a Hard Disk is very large as compared to floppy disk?

4.19 Describe various features of a Hard Disk?

4.20 What is a CD-ROM? How does it differ from Hard disk?

5

DATA REPRESENTATION

A general-purpose digital computer can perform a wide variety of data-processing jobs. It follows a sequence of instructions, called a program written in a computer language. Digital computers respond numbers rather than letters. These numerals have specific meaning for computers.

✦ Figure 5.1 shows a combination of ON and OFF bulbs Each bulb may represent digit 1 when ON and 0 when OFF. Sixteen combinations are possible by turning ON or OFF these four bulbs. Each combination may be used to convey some predefined message.

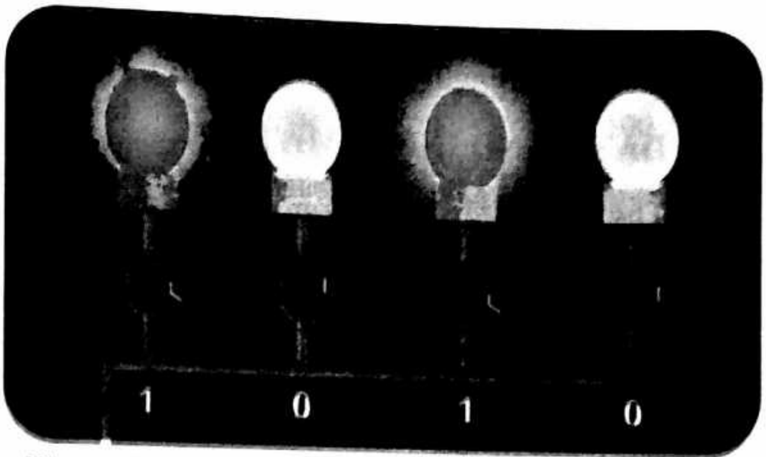


Figure 5.1 Combinations of ON/OFF bulbs may be used to convey a message.

Computers are often called data processors. Numbers, facts, names etc are represented by symbols. A symbol may be a printed character or a picture as shown in Figure 5.2. In abacus numbers are represented by the position and arrangement of beads. Every bead

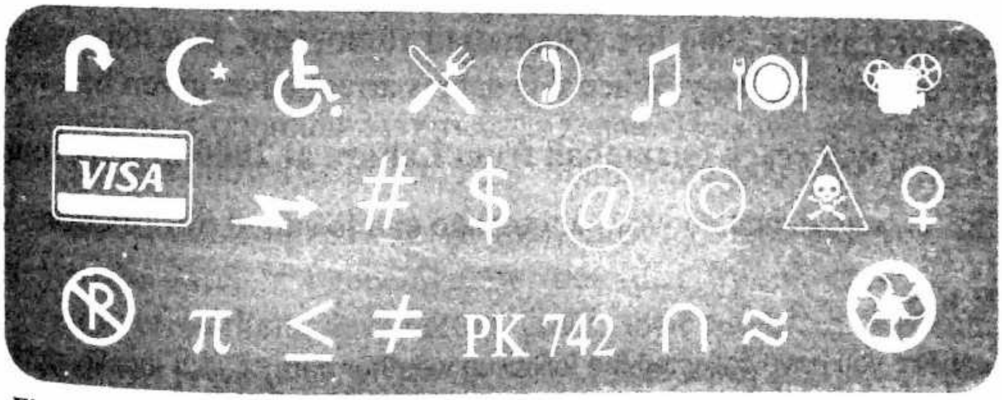


Figure 5.2 Some Symbols communicate some information among humans.

has a specific value such as 1, 2 …10, 20, 200, 3000 and so on. In this way, represent a particular number. In a computer anything that you type or draw, recognized by it in a numeric format. For example a word that appears on screen as:

PAKISTAN

is recognized by the computer in the form of a series of a group 0's and 1's shown below. Each group is in the binary format and represents one character.

01010000 01000001 01001011 01001001 01010011 01010100 01000001 0100111

Digital computers are built from electronic components with two state like ON and OFF states of switches. These two states 0 and 1 digits. As binar number system also uses two digits 0 and 1, therefore binary number system used in electronic computers. Machine language which a computer understands based on the combination 0's and 1's.

5.1 DATA AND INFORMATION

The computers work on data. The word data and information are ofte used interchangeably. There is an important distinction between them. Data is th words, numbers and graphics that describe people, events, things or ideas. Dat becomes information when it is arranged in a form that is meaningful.

A representation of facts, concepts or instructions suitable fo communication, interpretation or processing by humans or machines is calle Data. In short 'DATA' mean **collection of raw facts**. When these facts ar arranged in a suitable manner they provide some **information**. For exampl **Naveed reach Lahore 2000 hrs. Islamabad tomorrow,** is a data which can b arranged in a number of way. When this data is arranged as given below:-

Naveed will reach Islamabad from Lahore at 2000 hrs. by tomorrow.

This arranged data conveys a message and is called information.

EXAMPLE 5.1

Suppose you have collected the following weather data of your locality during a week On Monday and Friday minimum temperature was 9°C and maximum temperature was 16°C It was rainfall on Wednesday, the maximum and minimum temperature was 14° C and 7°C

respectively and the same was observed on Thursday. The temperature on Saturday was 10°C minimum and 18°C maximum and on Sunday it was 9°C minimum and 20°C maximum. It was cloudy from Monday to Thursday. The description of these facts is called data. These facts may be arranged in a suitable manner as shown in Table 5.1, and is called information.

TABLE 5.1 A Weather Chart

Day	Temperature		Rain/Cloudy/ Sunshine
	Minimum	Maximum	
Saturday	10°C	21°C	Sunshine
Sunday	9°C	20°C	Sunshine
Monday	9°C	16°C	Cloudy
Tuesday	8°C	15°C	Cloudy
Wednesday	7°C	14°C	Rain
Thursday	7°C	14°C	Cloudy
Friday	9°C	16°C	Sunshine

EXAMPLE 5.2

Let us consider another example. In a class test Hamid obtained 53 marks out of 100 marks in Urdu, 75 marks out of 100 marks in mathematics, 69 marks out of 100 marks in Physics, 62 marks out of 100 marks in Chemistry and 47 marks out of 100 marks in English. Aslam in the same test obtained 88 marks in Mathematics, 61 marks in Urdu, 80 marks in Physics, 52 marks in English and 58 marks in Chemistry. Rehmat scored 45 marks in English, 44 marks in Chemistry, 49 marks in Urdu 51 marks in Physics and 62 marks in Mathematics.

The above facts form the data. It may be processed and arranged as shown in table 5.2 to provide information based on given data. In this table marks of individual students are added and their percentage is calculated to determine their grades. Grade A is given to those scoring 60% and above while grade B is given to those getting 50% and above but less than 60%.

TABLE 5.2 RESULT OF STUDENTS

Name of Student	Marks Obtained in					Total	Percentage	Grade
	Urdu	Eng.	Math.	Phy.	Chem.			
HAMID	53	47	75	69	62	306	61.2 %	A
ASLAM	61	52	88	80	58	339	67.8 %	A
REHMAT	49	45	62	51	44	251	50.2 %	B

Information is obtained by processing data. Thus conversion of raw facts into a usable form is known as information. Some more examples of data are:-

i) Number of students (boys & girls) in various classes of a school.
ii) Various items in a general store, their quantity and rates.
iii) Name of workers, designation, grade, their addresses, salaries etc.

5.2 TYPES OF DATA

As we know that **data** is the collection of raw facts, therefore, we need symbols for their representation. These symbols may be letters, words or figures

such as 3 books, 5 students, 23rd March, 40 WATTS, PK543. Each data i conveys certain meanings. In computers data items can be classified into following three types.

Numeric Data. Alphabetic Data. Alphanumeric Data

NUMERIC DATA

Numeric data contains discrete numbers only such as 7, 505, 3.024, –12.6, etc. Numeric data can be integer data or real data.

INTEGER DATA

Integer data consists of positive and or negative whole numbers includi zero. For example +3, +7, –26 are integers.

REAL DATA

Real data consists of numbers which may be fractions or incremen including integer numbers such as 15.4, , 0.14 etc. are all real numbers.

ALPHABETIC DATA

Alphabetic data includes all the uppercase and lowercase letters alphabets and combination such as A, B, …Z, a, b, … z, NBP, Naveed, book etc

ALPHANUMERIC DATA

Alphanumeric data contains a combination of numerals and letters alphabets including special characters such as ? #, %, *, etc. For example 137 B/489, PK 635, 4A, F-16, MARCH 97 etc.

5.3 NUMBER SYSTEM

Digital computers are the machines which respond numbers. All the da and instructions must therefore be represented in a numeral format. Thus numb systems are very important to understand because a computer understan numbers only. The binary number system has been found to be the most natur and efficient system for modern digital machines.

5.4 THE DECIMAL NUMBER SYSTEM

The decimal number system consists of 10 digits from 0 to 9. The value each digit in a number depends upon the following:-

- The face value of the digit, i.e. the digit itself.
- The base of the system.
- The position of the digit in the number

such as 3 books, 5 students, 23rd March, 40 WATTS, PK543. Each data conveys certain meanings. In computers data items can be classified into following three types.

Numeric Data. Alphabetic Data. Alphanumeric Data

NUMERIC DATA

Numeric data contains discrete numbers only such as 7, 505, 3.024, –12.6, etc. Numeric data can be integer data or real data.

INTEGER DATA

Integer data consists of positive and or negative whole numbers including zero. For example +3, +7, –26 are integers.

REAL DATA

Real data consists of numbers which may be fractions or increments including integer numbers such as 15.4, , 0.14 etc. are all real numbers.

ALPHABETIC DATA

Alphabetic data includes all the uppercase and lowercase letters alphabets and combination such as A, B, …Z, a, b, … z, NBP, Naveed, book etc.

ALPHANUMERIC DATA

Alphanumeric data contains a combination of numerals and letters alphabets including special characters such as ? #, %, *, etc. For example 13 B/489, PK 635, 4A, F-16, MARCH 97 etc.

5.3 NUMBER SYSTEM

Digital computers are the machines which respond numbers. All the data and instructions must therefore be represented in a numeral format. Thus number systems are very important to understand because a computer understands numbers only. The binary number system has been found to be the most natural and efficient system for modern digital machines.

5.4 THE DECIMAL NUMBER SYSTEM

The decimal number system consists of 10 digits from 0 to 9. The value of each digit in a number depends upon the following:-

- The face value of the digit, i.e. the digit itself.
- The base of the system.
- The position of the digit in the number

EXAMPLE 5.3

Let us consider a decimal number 583. the positions of digits 3, 8 and 5 are at unit, tens and at hundredth places respectively. Thus 583 can be written as

SOLUTION: Hundreds (10^2) Tens (10^1) Units (10^0) ← (Position Value)

$$583 = 5 \quad 8 \quad 3$$
$$= 500 + 80 + 3$$
$$= 5 \times 100 + 8 \times 10 + 3 \times 1$$
$$\text{or } 583 = 5 \times 10^2 + 8 \times 10^1 + 3 \times 10^0 \qquad (\text{since } 1 = 10^0)$$

5.5 THE BINARY NUMBER SYSTEM

The ON/OFF positions of electronic switches in a digital computer correspond to **binary digits** 0 or 1. In binary number system any number can be expressed by the digits 0 and 1 only. The place or position value of each digit, in a binary number, is twice the place value of the digit on its right.

In binary system	1	corresponds to		$2^0 = 1$	in decimal system
" " "	10	"	"	$2^1 = 2$	" " "
" " "	100	"	"	$2^2 = 4$	" " "
" " "	1000	"	"	$2^3 = 8$	" " "

and so on.

EXAMPLE 5.4 Convert binary number $(101011)_2$ into its decimal equivalent.

SOLUTION:

	2^5	2^4	2^3	2^2	2^1	2^0	
	32	16	8	4	2	1	← (Place Value)
$(101011)_2 =$	1	0	1	0	1	1	

$$\text{Or } (101011)_2 = 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$$
$$\text{Or } (101011)_2 = 32 + 0 + 8 + 0 + 2 + 1$$
$$\therefore (101011)_2 = (43)_{10}$$

5.6 THE OCTAL NUMBER SYSTEM

The Octal Number System consists of 8 digits 0, 1, 2, 3, 4, 5, 6 and 7. In this number system, the base is 8. Each digit position in octal number system represents a power of eight. The numbers next to 7 will be 10, 11 … 17, 20, 21 … 27, 30, 31 … and so on.

EXAMPLE 5.5 Convert an Octal number $(236)_8$ into its decimal equivalent.

SOLUTION: 8^2 (=64) 8^1 (=8) 8^0 (=1) ← (Position Value)

$$\text{Since } (236)_8 = 2 \quad 3 \quad 6$$
$$= 2 \times 64 + 3 \times 8 + 6 \times 1$$
$$= 128 + 24 + 6$$
$$\therefore (236)_8 = (158)_{10}$$

We can convert an octal number into binary number by replacing each digit of octal number into its binary equivalent in a 3-digit format as shown below in the table.

OCTAL NUMBER	BINARY EQUIVALENT (Group of 3 digits)
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

EXAMPLE 5.6 Change the Octal number $(236)_8$ into its Octal equivalent.

SOLUTION: Starting from leftmost digit, 2, 3 and 6 numbers have binary equivalent in three digits format as, (010), (011) and (110) respectively.

Thus $(236)_8 = (010 \quad 011 \quad 110)_2$

EXAMPLE 5.7 Change binary number $(100111001101)_2$ into Octal number.

SOLUTION:

$(10111100101)_2 = 010 \quad 111 \quad 100 \quad 101$ **(Groups each having 3 binary digits)**

$= 2 \quad 7 \quad 4 \quad 5$ **(Equivalent octal digit for each binary group)**

Thus $(100111001101)_2$ is equivalent to $(2745)_8$ in octal system.

5.7 THE HEXADECIMAL NUMBER SYSTEM

The Hexadecimal number system consists of 16 digits 0, 1, … 8, 9, A, B, C, D, E and F. The alphabets A, B, C, D, E and F are used to represent decimal numbers 10, 11, 12, 13, 14 and 15 respectively. In this number system the base is 16. The next numbers after F will be 10, 11, … 19, 1A, 1B, … 1F, and so on.

CONVERSION OF HEXADECIMAL NUMBER INTO BINARY

EXAMPLE 5.8 Change hexadecimal number $(5C7)_{16}$ into its decimal equivalent.

SOLUTION: 16^2 **(=256)** 16^1 **(=16)** 16^0 **(=1)**

$$
\begin{aligned}
\text{Since } (5C7)_{16} &= 5 \qquad C \qquad 7 \\
&= 5 \times 256 + 12 \times 16 + 7 \times 1 \\
&= 1280 + 192 + 7 \\
\therefore \quad (5C7)_{16} &= 1479
\end{aligned}
$$

CONVERSION OF BINARY NUMBER INTO HEXADECIMAL

Conversion of a binary number into Hexadecimal number is very simple. Start breaking up the binary number from right into 4-bits group, each having four binary digits. Next assigning equivalent hexadecimal digit to each group.

EXAMPLE 5.9 convert binary number 10010011101 into hexadecimal number.

SOLUTION:

As $(10010011101)_2$ = 0100 1001 1101 in binary system

$\therefore$ $(10010011101)_2$ = 4 9 D in Hexadecimal system

Thus **$(10010011101)_2$** in Hexadecimal number will be **49D**

Table 5.4 4-bit group of Binary equivalent for each Hexadecimal digit.

Hexadecimal Number	Binary Equivalent (4-bit group)	Hexadecimal Number	Binary Equivalent (4-bit group)
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Table 5.5 below shows decimal numbers from 0 to 20 and their equivalent numbers in Binary, Octal and Hexadecimal system.

TABLE 5.5 Binary Octal and Hexadecimal equivalent for Decimal Numbers

Decimal Number	Binary equivalent	Octal Equivalent	Hexadecimal equivalent
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

5.8 NUMBER CONVERSION

We in general use decimal number system in our daily life. However, binary number system being suitable is used in digital computers. Other number systems are used for coding and storage purposes. Thus inter-conversion of numbers from one number system into other number system becomes important for computation and data processing in digital computers.

BINARY TO DECIMAL CONVERSION

To convert a binary number into its decimal equivalent, add the decimal equivalent of each position occupied by all the 1's in the binary number.

EXAMPLE 5.10 Convert binary number 10110 into decimal number.

SOLUTION:

	16	8	4	2	1	← (Position Value)
$(10110)_2 =$	1	0	1	1	0	

$$= 1\times16 + 0\times8 + 1\times4 + 1\times2 + 0\times1$$
$$= 16 + 0 + 4 + 2 + 0$$

$\therefore\ (10110)_2 = (22)_{10}$

DECIMAL TO BINARY CONVERSION

To convert a decimal integer into its binary equivalent, progressively divide the decimal number by 2, noting down the remainder taken in reverse order to form the binary equivalent.

EXAMPLE 5.11 Find the binary equivalent of decimal number 50.

SOLUTION:

2	50	
2	25	— 0
2	12	— 1
2	6	— 0
2	3	— 0
2	1	— 1
	0	— 1

1 1 0 0 1 0

Hence $(50)_{10} = (1\,1\,0\,0\,1\,0)_2$

OCTAL TO DECIMAL CONVERSION

To convert an octal number into its decimal equivalent, multiply the fac value of each digit by its position value to find the decimal equivalent of eac digit, their sum will be the decimal equivalent of the given octal number.

EXAMPLE 5.12 Convert octal number 3057 into its equivalent decimal numbe

SOLUTION:

	8^3 (=512)	8^2 (=64)	8^1 (=8)	8^0 (=1)	← (Position Value
$(3057)_8 =$	3	0	5	7	

$$= 3\times512 + 0\times64 + 5\times8 + 7\times1$$
$$= 1536 + 0 + 40 + 7$$

$\therefore\ (3057)_8 = (1583)_{10}$

An alternate method to convert an octal number into its decimal equivalent is give on next page.

ALTERNATE METHOD

$$
\begin{array}{rcccc}
 & 3 & 0 & 5 & 7 \\
 & \times 8 & & & \\
\hline
 & 24 & & & \\
 & +0 & & & \\
\hline
 & 24 & & & \\
 & \times 8 & & & \\
\hline
 & 192 & & & \\
 & +5 & & & \\
\hline
 & 197 & & & \\
 & \times 8 & & & \\
\hline
 & 1576 & & & \\
 & +7 & & & \\
\hline
 & 1583 & & &
\end{array}
$$

Thus $(3057)_8 = (1583)_{10}$

DECIMAL TO OCTAL CONVERSION

To convert decimal integer to its octal equivalent, progressively divide the decimal number by 8 and note down the remainder in reverse order to obtained its octal equivalent.

EXAMPLE 5.13 Convert decimal number 1583 into its Octal equivalent.

SOLUTION:

$$
\begin{array}{r|l}
8 & 1583 \\
\hline
8 & 197 - 7 \\
\hline
8 & 24 - 5 \\
\hline
8 & 3 - 0 \\
\hline
 & 0 - 3
\end{array}
\qquad 3 \quad 0 \quad 5 \quad 7
$$

Hence $(1583)_{10} = (3057)_8$

HEXADECIMAL TO DECIMAL CONVERSION

To convert a Hexadecimal number into its decimal equivalent, multiply the face value of each digit to it's position value to find the decimal equivalent of each digit, their sum will be the decimal equivalent of the given Hexadecimal number.

EXAMPLE 5.14 Convert hexadecimal number 3F8A into its decimal equivalent.

SOLUTION:

$$
\begin{array}{rcccc}
 & 16^3\,(=4096) & 16^2\,(=256) & 16^1(=16) & 16^0(=1) \\
(3F8A)_{16} = & 3 & F & 8 & A
\end{array}
$$

$$= 3 \times 4096 + 15 \times 256 + 8 \times 16 + 10 \times 1$$

$$= 12288 + 3840 + 128 + 10$$

$\therefore (3F8A)_{16} = 16266$

An alternate method for the conversion of Hexadecimal number into decimal number is described on the next page.

ALTERNATE METHOD

$$
\begin{array}{r l}
 & 3 \quad F \quad 8 \quad A \\
 & \times 16 \\
\hline
 & 48 \\
 & +15 \leftarrow F \\
\hline
 & 63 \\
 & \times 16 \\
\hline
 & 1008 \\
 & +8 \leftarrow 8 \\
\hline
 & 1016 \\
 & \times 16 \\
\hline
 & 16256 \\
 & +10 \leftarrow A \\
\hline
 & 16266
\end{array}
$$

Thus $(3F8A)_{16} = (16266)_{10}$

DECIMAL TO HEXADECIMAL CONVERSION

To convert decimal integer into its Hexadecimal equivalent, progressively divide the decimal number by 16 and note down the remainder in reverse order to form its hexadecimal equivalent.

EXAMPLE 5.15 Convert decimal number 20154 into its hexadecimal equivalent.

SOLUTION:

$$
\begin{array}{r|r l}
16 & 20154 & \\
\hline
16 & 1259 - 10 & = A \\
\hline
16 & 78 - 11 & = B \\
\hline
16 & 4 - 14 & = E \\
\hline
 & 0 - 4 &
\end{array}
\qquad 4 \quad E \quad B \quad A
$$

Hence $(20154)_{10} = (4EBA)_{16}$

5.9 REPRESENTING OF NUMBERS USING COMPLEMENTS

In decimal number system, complement of a number N is another number N such that their sum (N + N) is equal to 10^n; where exponent n is the number of digits in the number N. For example complement of 2 is 8, and that of 4 is 6.

EXAMPLE 5.16 Find the complements of (i) 25 (ii) 468 and (iii) 3097

SOLUTION:

i) Complement of 25 $= 10^2 - 25 = 75$ $\qquad 25 + 75 = 100$

ii) Complement of 468 $= 10^3 - 468 = 532$ $\qquad 468 + 532 = 1000$

iii) Complement of 3097 $= 10^3 - 3097 = 6903$ $\qquad 3097 + 6903 = 10000$

The above complements are also referred as **10 s complements**. In a similar manner **9 s complement** of a number N is another number N such that their sum is equal to one less than 10^n where exponent n is the number of digits in the number N. For example 9 s complement of 2 is 7, and that of 64 is 35.

EXAMPLE 5.17 Find the 9 s complements of (i) 25 (ii) 468 and (iii) 3097

SOLUTION:

i) 9 s Complement of 25 $= (10^2-1) - 25 = 74 \because 25 + 74 = 99$

ii) 9 s Complement of 468 $= (10^3-1) - 468 = 531 \because 468 + 531 = 999$

iii) 9 s Complement of 3097 $= (10^4-1) - 3097 = 6902 \because 3097 + 6902 = 9999$

COMPLEMENT OF A BINARY NUMBER

In modern digital computers binary number system is used. The binary numbers have either 2 s complement or 1 s complement.

2'S COMPLEMENT OF A BINARY NUMBER

2's complement of a binary number can be found by subtracting the number from 2^n where n is the number of digits in the binary number.

EXAMPLE 5.18 Find 2 s complement of (i) 101 (ii) 10110 (iii) 10101

SOLUTION: (i) 2 s complement of 101 $= 2^3 - 101$
$= 1000 - 101$
$= 011$

(ii) 2 s complement of 10110 $= 2^5 - 10110$
$= 100000 - 10110$
$= 01010$

(iii) 2 s complement of 10101 $= 2^5 - 10101$
$= 100000 - 10101$
$= 01011$

1 S COMPLEMENT OF A BINARY NUMBER

1 s complement of a binary number is 1 less than its 2 s complement. It can be found by subtracting the binary number from $(2^n - 1)$ where n is the number of digits in the given binary number.

EXAMPLE 5.19 Find 1 s complement of the following binary numbers:-
(i) 101 (ii) 10110 (iii) 10101

SOLUTION: (i) 1 s complement of 101 $= (2^3 - 1) - 101$
$= (1000 - 1) - 101$
$= 111 - 101$
$= 010$

(ii) 1 s complement of 10110 $= (2^5 - 1) - 10110$
$= (100000 - 1) - 10110$
$= 11111 - 10110$
$= 01001$

(iii) 1 s complement of 10101 $= (2^5 - 1) - 10001$
$= (100000 - 1) - 10101$
$= 11111 - 10101$
$= 01010$

NOTE: 1 s complement of a binary number can be obtained on replacing all its 0 s by 1 s and 1 s by 0 s. For 2 s complement of a binary number, add 1 in its 1's complement.

EXAMPLE 5.20 Find 1 s and 2 s complement of the following binary numbers:.

(i) 10 (ii) 1010 (iii) 10001

SOLUTION: (i) 1's complement of 10 = 01 (replacing 0 s by 1 s and 1 s by 0 s)
2's complement of 01 = 01+1 (adding 1 in 1's complement)
= 10

(ii) 1's complement of 1010 = 0101 (replacing 0 s by 1 s and 1 s by 0 s)
2's complement of 1010 = 0101+1 (adding 1 in 1's complement)
= 0110

(iii) 1's complement of 10001 = 01110 (replacing 0 s by 1 s and 1 s by 0 s)
2's complement of 10001 = 01110 +1 (adding 1 in 1's complement)
= 01111

5.10 BINARY ARITHMETIC

One of the major advantages of the binary number system is the ease with which arithmetic operations can be performed in digital computers. Very few rules and no tables is required to remember to solve any addition, subtraction, division multiplication or problem.

ADDITION

Addition has the following four rules.

(i) $0 + 0 = 0$ **(ii)** $0 + 1 = 1$ **(iii)** $1 + 0 = 1$ **(iv)** $1 + 1 = 1$

EXAMPLE 5.21 Add the binary numbers 1101 and 101.

SOLUTION: **Binary System** **Decimal Equivalent**

		1101			13
Adding 101	+	101	Adding decimal equivalent	+	5
Answer		10010	Answer		18

EXAMPLE 5.22 Add the binary numbers 11001 and 1110.

SOLUTION: **Binary System** **Decimal Equivalent**

		11001			25
Adding 1110	+	1110	Adding decimal equivalent	+	14
Answer		100111	Answer		39

EXAMPLE 5.23 Add the binary numbers 10110, 1011 and 1101

SOLUTION:

	Binary System		Decimal Equivalent
	10110		22
Adding 1011	+ 1011	Adding decimal equivalent	+ 11
	100001		33
Adding 1101	+ 1101	Adding decimal equivalent	+ 13
Answer	101110	Answer	46

SUBTRACTION

Subtraction of binary numbers is carried out by complement's method either by 2 s complement or 1 s complements.

EXAMPLE 5.24 Subtract the following numbers in binary system.
(i) $(1101)_2$ from $(10010)_2$ and (ii) $(10001)_2$ from $(11010)_2$

2 s Complement Method

(i) Binary number 1101 = 01101 (Adding 0's on the left to make equal digits in both the numbers)

2 s complement of 01101 = 10011.

	10010
Adding 2 s complement of 1101	+ 10011
Dropping (or ignoring) the carry	1 00101
Answer	00101

Thus $(10010)_2 - (1101)_2 = (101)_2$

(ii) 2 s complement of 10001 =01111

	11010
Adding 2 s complement of 10001	+ 01111
Dropping (ignoring) the carry	1 01001
Answer	01001

Thus $(11010)_2 - (10001)_2 = (1001)_2$

1 s Complement Method

(i) Binary number 1101 = 01101 (Adding 0's on the left to make equal digits in both the numbers))

1 s complement of 01101 = 10010

	10010
Adding 1 s complement of 1101	+10010
	1 00100
Adding the carry	+1
Answer	00101

Thus $(10010)_2 - (1101)_2 = (101)_2$

EXAMPLE 5.23 Add the binary numbers 10110, 1011 and 1101

SOLUTION:

		Binary System			Decimal Equivalent
		10110			22
Adding 1011	+	1011	Adding decimal equivalent	+	11
		100001			33
Adding 1101	+	1101	Adding decimal equivalent	+	13
Answer		101110	Answer		46

SUBTRACTION

Subtraction of binary numbers is carried out by complement's method either by 2 s complement or 1 s complements.

EXAMPLE 5.24 Subtract the following numbers in binary system.

(i) $(1101)_2$ from $(10010)_2$ and (ii) $(10001)_2$ from $(11010)_2$

2 s Complement Method

(i) Binary number 1101 = 01101 (Adding 0's on the left to make equal digits in both the numbers)

2 s complement of 01101 = 10011.

	10010
Adding 2 s complement of 1101	+ 10011
Dropping (or ignoring) the carry	① 00101
Answer	00101

Thus $(10010)_2 - (1101)_2 = (101)_2$

(ii) 2 s complement of 10001 =01111

	11010
Adding 2 s complement of 10001	+ 01111
Dropping (ignoring) the carry	① 01001
Answer	01001

Thus $(11010)_2 - (10001)_2 = (1001)_2$

1 s Complement Method

(i) Binary number 1101 = 01101 (Adding 0's on the left to make equal digits in both the numbers))

1 s complement of 01101 = 10010

	10010
Adding 1 s complement of 1101	+10010
	①00100
Adding the carry	+1
Answer	00101

Thus $(10010)_2 - (1101)_2 = (101)_2$

(ii) 1's complement of 10001 = 01110

$$\begin{array}{rr}
 & 11010 \\
\text{Adding 1's complement of 10001} & +\ 01110 \\
\hline
 & \textcircled{1}\ 01000 \\
\text{Adding the carry} & \hookrightarrow\quad +1 \\
\hline
\text{Answer} & 01001 \\
\hline
\end{array}$$

Thus $(11010)_2 - (10001)_2 = (1001)_2$

MULTIPLICATION

Multiplication has the following four rules.

(i) $\begin{array}{r} 0 \\ \times\ 0 \\ \hline 0 \\ \hline \end{array}$ **(ii)** $\begin{array}{r} 0 \\ \times\ 1 \\ \hline 0 \\ \hline \end{array}$ **(iii)** $\begin{array}{r} 1 \\ \times\ 0 \\ \hline \times\ \times \\ \hline \end{array}$ **(iv)** $\begin{array}{r} 1 \\ \times\ 1 \\ \hline 1 \\ \hline \end{array}$

The above rules can be reduced to only two rules. That is only $1 \times 1 = 1$ and for all the remaining combinations, the result will be zero (0).

EXAMPLE 5.25 multiply the binary numbers 1101 by 101

SOLUTION:

$$\begin{array}{rl}
\underline{\textbf{Binary Number}} & \\
1\ 1\ 0\ 1 & \\
\times\quad 1\ 0\ 1 & \\
\hline
1\ 1\ 0\ 1 & \leftarrow \text{Partial product} \\
0\ 0\ 0\ 0\ \times & \leftarrow \text{Partial product} \\
1\ 1\ 0\ 1\ \times\ \times & \leftarrow \text{Partial product} \\
\hline
1\ 0\ 0\ 0\ 0\ 0\ 1 & \leftarrow \text{Final product} \\
\hline\hline
\end{array}
\qquad
\begin{array}{r}
\underline{\textbf{Decimal Equivalent}} \\
1\ 3 \\
\times\quad 5 \\
\hline
6\ 5 \\
\hline\hline
\end{array}$$

Binary Multiplication is a series of additions. It needs no tables and no carries (except in adding partial products)

EXAMPLE 5.26 Multiply the numbers 10001 by 1100 in binary system.

SOLUTION:

$$\begin{array}{rl}
\underline{\textbf{Binary Number}} & \\
1\ 0\ 0\ 0\ 1 & \\
\times\quad 1\ 1\ 0\ 0 & \\
\hline
1\ 0\ 0\ 0\ 1\ \times\ \times & \leftarrow \text{Partial product} \rightarrow \\
1\ 0\ 0\ 0\ 1\ \times\ \times\ \times & \leftarrow \text{Partial product} \rightarrow \\
\hline
1\ 0\ 0\ 0\ 1\ 1\ 0\ 0 & \leftarrow \text{Final product} \rightarrow \\
\hline\hline
\end{array}
\qquad
\begin{array}{r}
\underline{\textbf{Decimal Equivalent}} \\
1\ 7 \\
\times\ 1\ 2 \\
\hline
3\ 4 \\
1\ 7\ \times \\
\hline
2\ 0\ 4 \\
\hline\hline
\end{array}$$

In the above example multiplication by 0's is omitted being unnecessary instead position value is shifted towards left as many places as many 0's occur.

DIVISION

Division can also be done in the usual way. By long division, the divisor is subtracted from the dividend and a 1 is placed in the quotient starting from the left. If the divisor cannot be subtracted, then a 0 is placed in the quotient.

EXAMPLE 5.27 Divide the number $(11101)_2$ by $(101)_2$ binay system.

SOLUTION:

```
                        1 0 1            ←Quotient
Divisor →   1 0 1 )  1 1 1 0 1           ←Dividend
                     1 0 1
                     -----
                       1 0 0 1
                         1 0 1
                       -------
                         1 0 0           ←Remainder
```

Thus $(101)_2$ divides $(11101)_2$ by $(101)_2$ times with a remainder of $(100)_2$.

EXAMPLE 5.28 Divide $(11101010)_2$ by $(101)_2$

SOLUTION:

```
                          1 0 1 0 1              ←Quotient
Divisor →   1 0 1 1 )  1 1 1 0 1 0 1 0           ←Dividend
                       1 0 1 1
                       -------
                           1 1 1 0
                           1 0 1 1
                           -------
                               1 1 1 0
                               1 0 1 1
                               -------
                                   1 1       ←Remainder
```

Thus $(1011)_2$ divides $(11101010)_2$ by $(10101)_2$ times with a remainder of $(11)_2$.

FIXED AND FLOATING POINT NUMBER REPRESENTATION

In mixed numbers the position of a point represents fractional part of a number. The position of a point in a mixed number can be represented in two ways by giving it a fixed position or by using a floating-point representation.

FIXED POINT REPRESENTATION

The usual way of representing numbers is to write the number with a decimal point fixed between two digits. For example: 2.7, 236.58 and 0.1864 etc.

FLOATING POINT REPRESENTATION

It is not always easy to express a number in fixed-point format such as 235000000, 0.00000047. This is particularly true in scientific computations. Such numbers are expressed using floating point representation. Floating-point representation has the following three components.

i) The mantissa also called the argument.
ii) The radix or base
iii) The exponent also called the characteristic

In this method a number is expressed as a combination of a mantissa an exponent. The mantissa is kept less than 1 but greater than or equal to 0.1 illustrated in the examples given in the table below.

Table 5.6 Showing decimal numbers in Floating point representation

Fixed point Representation	Floating point Representation	
	Scientific Version	Computerized Version
435.0	0.435×10^{3}	+ 0.435 E +3
43500.0	0.435×10^{5}	+ 0.435 E +5
0.00435	0.435×10^{-2}	+ 0.435 E –2
29580.2	0.295802×10^{5}	+ 0.95802 E +5
0.000066	0.66×10^{-4}	+ 0.66 E –4

In case of binary numbers, the radix takes the value 2 for the exponent. Table 5.7. below illustrates binary numbers in floating point format.

Table 5.7 Showing binary numbers in Floating point representation

Fixed point Representation	Floating Point Representation		
	Scientific Version	Computerized Version	
		Fraction	Exponent
$(11)_2$	0.11×2^{2}	01100000	010
$(101.101)_2$	0.101101×2^{3}	01011010	011
$(0.001011)_2$	0.1011×2^{-2}	01011000	- 010
$(1101.11)_2$	0.110111×2^{4}	01101110	100
$(11011.01)_2$	0.1101101×2^{5}	01101101	101

In the memory location only the mantissa and the exponents are stored. Radix always assumes the same value $\{(10)_2 = 2$ for binary$\}$ which is not stored. Binary numbers shown in Table 5.7 can be represented in a 16-bit register using 8-bits for the mantissa and the exponent while the remaining two bits are used as sign bit each for mantissa and the exponent. For positive numbers sign-bit is 0 and for negative number sign bit is 1 as shown below in Table 5.8.

Table 5.8 Showing binary numbers with sign bit in Floating point representation

Floating-point representation in Scientific Version	Floating-point representation in 16-bit register															
	SIGN BIT	Mantissa								SIGN BIT	Exponent					
0.11×2^{2}	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0
-0.101101×2^{3}	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	1
0.1011×2^{-2}	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1	0
0.110111×2^{4}	0	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0
0.1101101×2^{5}	0	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1

Floating-point representation makes the computation simpler and also increases the range of numbers that can be stored in a given memory location.

5.11 CODING SCHEME

A single binary digit is called a 'bit'. All information in a digital system is represented in the group of bits arranged in sequence. The number of bits in the data sequences processed by a given computer is called its word size. An 8-bit microprocessor can retrieve, process, store and transmit data or instructions in 8-bit or one byte word. Similarly the word size of a 16-bit microprocessor is two bytes long. The language of electronic computers is binary therefore all the numeric or non-numeric data must be converted into binary language. This is called coding. Coding is the process of representing all the numeric or non-numeric information in binary digits..

5.12 CODES USED IN COMPUTERS

Digital computers use binary numbers for performing arithmetic operations on data. Computers read input number data and print answers in decimal numbers. Different types of codes are used to represent numerical data, one of which is binary-coded decimal (BCD).

BINARY-CODED DECIMAL (BCD)

In BCD code, each decimal digit is represented by its four-bit binary equivalent instead of converting the entire decimal number into its pure binary equivalent. For example the BCD code for a decimal number 948 will be as follows:

Decimal	9	4	8
BCD	1001	0100	1000

Hence the number $(948)_{10}$ in BCD code = $(1001\ 0100\ 1000)_2$

ALPHANUMERIC CODES

BCD code could represent only a maximum of $2^4 = 16$ combinations. when the data contains letters and symbols in addition to numbers then 6-bit BCD code was used. A 6-bit BCD code could provide $2^6 = 64$ possible combinations to represent 64 different characters. Soon it was realized that 6-bit code are not sufficient to provide codes for numbers 0 to 9, small and capital letters of alphabets and other special characters for punctuation and mathematical operations. For this reason 6-bit BCD code was extended to 7 and 8-bit alphanumeric codes to represent 128 and 256 symbols respectively. One such standard code is a 7-bit American Standard Code for Information Interchange (ASCII) another one is Extended Binary coded Decimal Interchange Code (EBCDIC) used in IBM mainframes.

THE ASCII CODE

ASCII is a 7-bit code used to handle alphanumeric data having 128 different characters. It stands for American Standard Code for Information Interchange. It is the most popular coding system for PC's and data communication. This code allows manufacturers to standardize input/output devices such as keyboards, printers, visual display units, etc. An extension of ASCII code uses 8-bits called as ASCII-8 code with an extra eighth bit as a **parity** bit.

The EBCDIC Code

EBCDIC (Extended binary coded Decimal Interchange Code) is an 8-bit code and can provide 256 different characters. It is primarily used by International Business Machine on IBM mainframes and on other large computers. Table 5.9 shows a comparison of some character representations in ASCII 7- bit code and EBCDIC 8-bit code.

Table 5.9 ALPHANUMERIC CHARACTER CODES IN 7-bit ASCII AND 8-bit EBCDIC

Character	ASCII	EBCDIC
A	100 0001	1100 0001
B	100 0010	1100 0010
C	100 0011	1100 0011
D	100 0100	1100 0100
E	100 0101	1100 0101
F	100 0110	1100 0110
G	100 0111	1100 0111
H	100 1000	1100 1000
I	100 1001	1100 1001
J	100 1010	1101 0001
K	100 1011	1101 0010
L	100 1100	1101 0011
M	100 1101	1101 0100
N	100 1110	1101 0101
O	100 1111	1101 0110
P	101 0000	1101 0111
Q	101 0001	1101 1000
R	101 0010	1101 0001
S	101 0011	1110 0010
T	101 0100	1110 0011
U	101 0101	1110 0100
V	101 0110	1110 0101
W	101 0111	1110 0110
X	101 1000	1110 0111
Y	101 1001	1110 1000
Z	101 1010	1110 1001
0	011 0000	1111 0000
1	011 0001	1111 0001
2	011 0010	1111 0010
3	011 0011	1111 0011
4	011 0100	1111 0100
5	011 0101	1111 0101
6	011 0110	1111 0110
7	011 0111	1111 0111
8	011 1000	1111 1000
9	011 1001	1111 1001

SUMMARY

Digital computers respond numbers rather than letters. called data processors. Numbers, facts, names etc are represented by symbols. As binary number system also uses two digits 0 and 1, therefore binary number system is used in electronic computers.

DATA AND INFORMATION

The computers work on data. DATA' mean collection of raw facts. When data is arranged to become meaningful, it is called information.

TYPES OF DATA

In computers data items can be classified into the following three types. Numeric Data. Alphabetic Data and the Alphanumeric Data

NUMBER SYSTEM

Digital computers respōnd numbers. Therefore data and instructions must be represented in a numeral format. Various number systems used are decimal number system, binary number system, octal number system and hexadecimal number system. We, in general, use decimal number system in our daily life. However, due to the binary nature of electronic devices, binary number system has been found to be the most natural and efficient system for modern digital machines. Other number systems such as Octal and Hexadecimal number systems are used for coding and storage purposes. Moreover the conversion of a binary number into octal or hexadecimal number and vice versa is quite simple.

THE BINARY NUMBER SYSTEM

The ON/OFF positions of electronic switches in a digital computer correspond to **binary digits** 0 or 1. In binary number system any number can be expressed by the digits 0 and 1 only. The place or position value of each digit, in a binary number, is twice the place value of the digit on its right. A number in decimal number system can be converted into its binary equivalent.

NUMBER CONVERSION

Thus inter-conversion of numbers from one number system into other becomes important for computation and data processing in digital computers.

REPRESENTING NUMBER USING COMPLEMENTS

In binary number system 2's complement and 1's complement of a number N is obtained by subtracting the binary number N from 2^n and 2^n -1 respectively, where exponent n is the number of digits in the number N. To find 1's complement of a binary number, simply change all its 1's into 0's and 0's into 1's. while to find 2's complement of a binary number add 1 in its 1's

complement. A computer performs subtraction using the principle of addition, subtraction or complementing.

BINARY ARITHMETIC

In digital computers very few rules are required to perform addition, subtraction, division and multiplication.

FIXED AND FLOATING POINT NUMBER REPRESENTATION

Position of the decimal point in a mixed number can be represented in two ways:- Fixed Point Representation and the Floating Point Representation.

Fixed Point Representation

The decimal point is fixed in its correct position between the appropriate digits in Fixed Point Representation. For example: 2.7, 23.008, 0.1...

Floating Point Representation Floating Point Representation

Floating-point representation has the three components: the mantissa, radix or base and the exponent. Mantissa and the exponents are stored in the memory location but not the Radix which is 2 for binary. To indicate whether a number is positive or negative sign bit 0 or 1 is used respectively.

CODING SCHEME

A single binary digit is called a 'bit'. The number of bits in the data sequences processed by a given computer is called its word size. Coding is a process of representing numeric or non-numeric data and instructions in binary

Codes Used In Computers

Digital computers use binary numbers for performing arithmetic operations on data. Different types of codes are used to represent numerical data one of which is BCD (binary-coded decimal).

Alphanumeric Codes

Codes such as EBCDIC, ASCII, ANSI or UNICODE have been developed when the data contains letters and symbols in addition to numbers.

The ASCII Code

ASCII is a 7-bit code used to handle alphanumeric data. An extension of ASCII code uses 8-bits called as ASCII-8 code with an extra eighth bit as a **parity** bit.

The EBCDIC Code

EBCDIC (Extended binary coded Decimal Interchange Code) is an 8-bit code primarily used by International Business Machine (IBM) and IBM compatible computer systems.

EXERCISES

5.01 Complete the following statements

i) Data is a collection of ________.

ii) Data becomes information when ________ properly.

iii) Octal equivalent of binary number 1100010 is _____ .

iv) 2's complement of binary number 1100010 is _____ .

v) In floating point representation, mantises is kept less than ______ .

vi) ASCII stands for ________________________________.

5.02 Tick (✓) the following statements either True or False.

i) PK 345 is an example of Alphabetic data. True/False

ii) 128 is not an octal number. True/False

iii) In hexadecimal number F is equivalent to 16 in decimal number. True/False

iv) EBCDIC code is used in IBM and IBM compatible computers. True/False

v) Binary Coded Decimal (BCD) is a 10-bit code. True/False

5.03 Encircle one choice A, B, C or D in each case.

i) Binary coded decimal (BCD) expresses each decimal digit as:-
(A) binary digit (B) byte (C) nibble (D) word

ii) The number 1000 comes immediately after:-
(A) 900 (B) FFF (C) 887 (D) 499

iii) The radix of a computerized version of the number (+0.66 E – 4) is:-
(A) 2 (B) 4 (C) 6 (D) 8

iv) The number of possible combinations in a 7-bit code are:-
(A) 49 (B) 64 (C) 128 (D) 256

v) Which of the following is a 16 bit code?
(A) BCD (B) Unicode (C) ASCII (D) EBCDIC

5.04 **Match the items given in Column I with those given in Column II**

	Column I		Column II
i)	Mantissa	a)	Nibble
ii)	Radix	b)	Byte
iii)	Exponent	c)	Argument
iv)	BCD	d)	Base
v)	ASCII	e)	Characteristics

5.06 Describe the importance of binary number system in computers.

5.07 Convert the following decimal numbers into their binary equivalent

(i) 47 (ii) 167 (iii) 0.875 (iv) 21.625

5.08 Convert the following binary numbers into their decimal equivalent:-

(i) 101101 (ii) 1001110 (iii) 0.10111 (iv) 101.0101

5.09 Convert the following numbers in decimal number system into their octa equivalent:

(i) 43 (ii) 291 (iii) 168 (iv) 2010

5.10 Convert the following numbers in octal system into their decimal equivalent:

(i) 27 (ii) 372 (iii) 1101 (iv) 2345

5.11 Convert the following numbers in decimal number system into their Hexadecimal equivalent:

(i) 37 (ii) 141 (iii) 532 (iv) 702

5.12 Convert the following numbers in Hexadecimal number system into their decimal equivalent:

(i) 4F (ii) BA4C (iii) FFAD (iv) 29E

5.13 Subtract: (i) 1010_2 from 1101_2 (ii) 101001_2 from 110110_2
(iii) 1111_2 from 10101_2 (iv) 10001_2 from 11100_2
(v) 1010_2 from 1001001_2 using 1's and 2's complements

5.14 What is meant by coding in computers?

5.15 Describe ASCII code and EBCDIC code.

6

BOOLEAN ALGEBRA

In the previous chapter you have learned that digital computers use binary numbers. A digital computer is made up of various types of simple circuits arranged and grouped according to the rules of symbolic logic. George Boole (1806 – 1876), the founder of symbolic logic presented it as mathematical logic, which is now known as Boolean algebra. The significance of these developments was not realized at that time. Boolean algebra finds its application to problems in the form of statements having answers either true or false. The binary nature of it is linked with the computer development.

Boolean algebra uses algebraic notation to express logical relationship in the same way as in ordinary algebra to express mathematical relationships. As all the electronic switching devices/circuits have two discrete states that is **ON** and **OFF** states therefore the **two-valued Boolean algebra,** which deals with these switching circuits, is also called **switching algebra**.

6.1 ELEMENTS OF BOOLEAN ALGEBRA

In Boolean algebra an expression can be formed using binary variables, constants, symbols of logic operations and the parentheses.

BOOLEAN CONSTANTS

In Boolean algebra a set of constants has only two elements **0** or **1**. Thus a Boolean constant is either 0 if not 1 or is 1 if not 0.

BOOLEAN VARIABLES

The variables used in Boolean algebra can be represented by letters of the alphabet such as **A, B, C, … x, y, z** etc. Each variable must take one of the two values; **1** or **0**. These two values may be given different names such as **true** or **false**; **yes** or **no**; **high** or **low**; **up** or **down**; **ON** or **OFF** etc.

6.2 LOGICAL OPERATORS

In Boolean algebra there are three basic operators, one unary operator **NOT** and two binary operator **AND** and **OR** as shown in table 6.1.

TABLE 6.1 Showing Symbols Used to represent various Logical Operations

OPERATION	SYMBOL USED	COMMENTS
NOT	Prime (′) or Bar (‾)	Negation of the value.
OR	Plus (+) or Union (∪)	Logical addition.
AND	Dot (·) or Intersection (∩)	Logical multiplication.

NOT OPERATORS

NOT operation is a unary operation. It is the negation of a quantity on which it operates. NOT of a variable (let X) is represented by putting a bar over it. Let the result is expressed by Z such that,

$$Z = \overline{X}$$

and is read as Z is equal to NOT of X or complement of X. It means that Z is not what X is. It is illustrated in the truth table 6.2.

Truth Table 6.2 for **NOT** OPERATION

X	$Z = \overline{X}$
0	1
1	0

OR OPERATION

In Boolean algebra **OR** operation means **logical addition** and is represented by plus sign between two variables. For example

$$Z = X \textbf{ OR } Y = X + Y$$

It is read as Z is equal to X OR Y. The resulting variable Z assumes the value 0 only if all the variables are 0, otherwise Z will be 1 if one or more variables are 1. The operation **OR** is illustrated in Truth table 6.3. between the variables X and Y.

Truth Table 6.3 for **OR** OPERATION

X	Y	Z
0	0	0
0	1	1
0	0	1
1	1	1

AND OPERATION

In Boolean Algebra the operation **AND** means **logical multiplication** and is represented with or without a dot between the variables as

$$Z = X \textbf{ AND } Y = X \cdot Y = X\,Y$$

It is read as Z is equal to X **AND** Y. Thus Z is 1 only if **X** and **Y** both have value 1 otherwise Z will be 0 if either or both X and **Y** are 0 as illustrated in Truth table 6.4.

Truth Table 6.4 for **AND** OPERATION

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

6.3 BOOLEAN EXPRESSION

In Boolean algebra an expression is a logical statement which is either true or false. In Boolean expression, statements are represented by variables connected or operated by logical operators. Let us study the statements A, B, C, D, E and F.

	Statement	Logical value
A .	The Sun sets in the West.	True
B .	The Sun does not set in the West.	False
C .	I am not sleeping	True
D .	I am sleeping	False
E .	Pencil is bigger than you	False
F .	$18 < 25$	True

We can form Boolean expressions as given below from the above statements being true or false. **(Note that questions and exclamations are not included as they do not form logical statements).**

A = True D = False
B = False E = False
C = True F = True

Giving value 1 to the true expressions and 0 to the false one, we get

$A = 1$ $D = 0$
$B = 0$ $E = 0$
$C = 1$ $F = 1$

Consider statement B. It is the negation of statement A. Thus

$$B = \overline{A} \text{ (NOT of A)}$$
$$\text{But } \overline{A} = 0 \qquad (\because A = 1)$$
$$\therefore B = 0$$

Similarly statement D is the negation of statement C. Hence

$$D = \overline{C} \text{ (NOT of C)}$$
$$\text{But } \overline{C} = 0 \qquad (\because C = 1)$$
$$\therefore D = 0$$

We can also construct compound statements joining simple statements by some logical operator which can be written in Boolean expression such as

He is happy **AND** he is rich = True / False

The above expression will be true only when both the statements are true otherwise the expression will be false if any one of them is false.

Consider the following compound statement joining the two statements by the operator OR

He is hungry **OR** he is thirsty = True / False

This expression will be true if either of the two or both the statements are true. The expression will be false only when both of them are false.

OPERATOR'S PRECEDENCE

A Boolean expression can be evaluated using the following precedence that Boolean operators must follow:

i) Expressions must be scanned from **left to right**.
ii) **Parentheses** are evaluated first.
iii) **NOT** operations are performed after parentheses.
iv) **AND** operations are performed next to **NOT**.
v) **OR** operations are performed at the end.

6.4 IDENTITY ELEMENTS

The element **0** is an **identity element** with respect to binary operation **OR** (+), that is

$$0 + A = A$$

The element **1** is an **identity element** with respect to binary operation **AND** (·), that is

$$1 \cdot A = A$$

6.5 LAWS OF BOOLEAN ALGEBRA

These laws define the structure of Boolean Algebra. These consist of postulates at theorems of Boolean Algebra. The postulates are the basic axioms of the Boolean Algebra and need no proof.

6.6 POSTULATES OF BOOLEAN ALGEBRA

The postulates or axioms of Boolean algebra are used to deduce the properties and theorems of Boolean algebra. These axioms do not need any proof. Some of the basic axioms of Boolean algebra are:-

Axiom 1. (Existence of identity elements)

There exist elements A and B belonging to a set K. These elements have value either 0 of 1 such that

a) $A + 0 = A$ and
b) $A \cdot 1 = A$

Axiom 2. (Commutative Law)

The binary operators (+) and (.) are commutative over the elements A and B belonging to set K since

a) $A + B = B + A$ and
b) $A \cdot B = B \cdot A$

Axiom 3. (Associative Law)

If elements A, B and C belong to a set K then according to associative law

a) $A + (B + C) = (A + B) + C$ and
b) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

Axiom 4. (Distributive Law)

According to the distributive law

a) $A \cdot (B + C) = A \cdot B + A \cdot C$ and
b) $A + (B \cdot C) = (A + B) \cdot (A + C)$

Axiom 5. (Existence of inverse)

For every element A belonging to set K there exists an element $\overline{A}$ belonging to the same set K such that

a) $A + \overline{A} = 1$ and
b) $A \cdot \overline{A} = 0$

DUALITY PRINCIPLE

The axioms of Boolean algebra have been listed in pairs as part (a) and part (b) One part may be obtained from the other if the binary operators (+ and .) as well as the identity elements (0 and 1) are interchanged. This is an important property of Boolean algebra. This property is called the **duality principle**. In Boolean algebra the values of identity elements and the elements of a set K are either 1 or 0. When we apply duality principle, we interchange the operators OR (+) and AND (.) and replace 0's by 1's and 1's by 0's. For example if we apply duality principle on axiom 5 (a). We get

$$\overline{A} \cdot A = 0 \quad \text{or} \quad A \cdot \overline{A} = 0$$

Here A is replaced by $\overline{A}$, $\overline{A}$ by A, operator '+' is replaced by operator '·' and 1 by 0. The expression so obtained is axiom 5 (b).

6.7 TRUTH TABLE

A truth table is a table that shows the result of a Boolean expression for all the possible combinations of the values given to the variables related to the

operators in the expression. For example, the Truth Tables 6.8 for operati AND, OR and NOT given below clearly define these operators.

Truth Tables 6.8 FOR LOGICAL OPERATIONS AND, OR and NOT

AND OPERATION

A	B	$X = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR OPERATION

A	B	$X = A+B$
0	0	0
0	1	1
1	0	1
1	1	1

NOT OPERATION

A	$X = \overline{A}$
0	1
1	0

6.8 THEOREMS OF BOOLEAN ALGEBRA

Boolean algebra is useful in changing Boolean expression or tl corresponding logic circuits into different, functionally equivalent but simpl forms. This helps to design a logic circuit to realize a given set of combination functions using minimum number of gates.

There are some theorems of Boolean algebra that are of fundamenta importance. These are helpful in simplifying Boolean expressions. Thes theorems can be proved with the help of the truth table or postulates (axioms) o Boolean algebra described above. These theorems are:

THEOREM 1: (*Idempotent laws*)

(a) $A + A = A$ **(b)** $A \cdot A = A$

PROOF: (a) L.H.S. $= A + A$

$= (A + A) \cdot 1$ [by axiom 1 (b)]

$= (A + A) \cdot (A + \overline{A})$ [by axiom 5 (a)]

$= A + (A \cdot \overline{A})$ [by axiom 4 (b)]

$= A + 0$ [by axiom 5 (b)]

$= A$ [by axiom 1 (a)]

$=$ R.H.S.

(b) L.H.S. $= A \cdot A$

$= A \cdot A + 0$ [by axiom 1 (a)]

$= A \cdot A + A . \overline{A}$ [by axiom 5 (b)]

$= A \cdot (A + \overline{A})$ [by axiom 4 (a)]

$= A \cdot 1$ [by axiom 5 (a)]

$= A$ [by axiom 1 (b)]

$=$ R.H.S.

THEOREM 2: **(a)** $A + 1 = 1$ **(b)** $A \cdot 0 = 0$

PROOF: (a) L.H.S. $= A + (A + \overline{A})$ [by axiom 5 (a)]
$= (A + A) + \overline{A}$ [by axiom 3 (a)]
$= A + \overline{A}$ [by axiom 1 (a)]
$= 1$ [by axiom 5 (a)]
$=$ R.H.S.

(b) L.H.S. $= A \cdot 0$
$= A \cdot (A \cdot \overline{A})$ [by axiom 5 (b)]
$= (A \cdot A) \cdot \overline{A}$ [by axiom 3 (b)]
$= A \cdot \overline{A}$ [by axiom 1 (b)]
$= 0$ [by axiom 5 (b)]
$=$ R.H.S.

THEOREM 3: (*absorption Law*)

(a) $A + (A \cdot B) = A$ **(b)** $A . (A + B) = A$

PROOF: (a) L.H.S. $= A + (A \cdot B)$
$= (A \cdot 1) + (A \cdot B)$ [by axiom 1 (b)]
$= A \cdot (1 + B)$ [by axiom 4 (a)]
$= A \cdot 1$ [by theorem 2 (a)]
$= A$ [by axiom 1(b)]
$=$ R.H.S.

(b) L.H.S. $= A \cdot (A + B)$
$= (A + 0) \cdot (A + B)$ [by axiom 1 (a)]
$= A + (0 \cdot B)$ [by axiom 4 (b)]
$= A + 0$ [by theorem 2 (b)]
$= A$ [by axiom 1 (a)]
$=$ R.H.S.

THEOREM 4: (*Involution Law*) $\overline{\overline{A}} = A$

It states that double complementation has cancellation effect. This can be proved by the method of **perfect induction** as shown in truth table 6.9. In this method a relation is checked by a truth table.

Truth Table 6.9

=

A	$\overline{A}$	$\overline{\overline{A}}$
0	1	0
1	0	1

THEOREM 5: (*Demorgan's Law*)

(a) $\overline{A + B} = \overline{A} \cdot \overline{B}$ (b) $\overline{A \cdot B} = \overline{A} + \overline{B}$

PROOF: (a) L.H.S. $= \overline{A + B}$

$= \overline{A} \;\bar{+}\; \overline{B}$ (by duality principle)

$= \overline{A} \cdot \overline{B}$

$=$ R.H.S.

(b) L.H.S. $= \overline{A \cdot B}$

$= \overline{A} \;\bar{\cdot}\; \overline{B}$ (by duality principle)

$= \overline{A} + \overline{B}$

$=$ R.H.S.

6.9 BOOLEAN FUNCTIONS

A Boolean function is an expression which can be formed by binary variables, the two binary operators {OR and AND}, one unary operator NOT, parentheses and equal sign. The function assumes the value either 0 or 1 for given values of the variables. Consider a Boolean function F formed by two variables A and B. Such that:

$$F = \overline{A} + A \cdot \overline{B}$$

The function is equal to 1 if A = 0 or if A = 1 and B = 0 otherwise F = 0 for all the remaining combination for the values of A and B · A Boolean function can be represented in a truth table. To represent function F in a truth table requires (2^2 =) 4 rows. Since there are two binary variables A and B and each can have value 0 and 1 and so four combination of A and B will be possible. In Boolean expression the literals A and B are assigned value 1 then their complements expressed as $\overline{A}$ (A bar) and $\overline{B}$ (B bar) are designated 0 value.

Truth Table 6.10 for $F = \overline{A} + A \cdot \overline{B}$

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

From the truth table 6.10 we find that there are three combinations of the binary variables for which F = 1 and only one combination for which F = 0. The value of the function F is 0 for both A = 1 and B = 1 The expression for the function F does not contain the term A · B, let us consider some more examples for function formed by there variables.

EXAMPLE 6.1 Construct truth table for the following Boolean expressions:

a) $A B \overline{C}$ b) $A + \overline{B} C$,

c) $A B C + A \overline{B} C$ d) $\overline{A}\,\overline{B} C + \overline{A} B C + A \overline{B}$

e) $A \overline{B} + \overline{A} C$

SOLUTION:

The value of function F_1 will be 1 when $A = 1$, $B = 1$ and $C = 0$; otherwise F_1 will be 0. A Boolean function can also be represented by a truth table. To represent a function in a truth table the number of rows in the table will be equal to 8 to represent 2^3 combinations for 3 binary variables each having two values 0 or 1. The column labeled F_1 contains either 0 or 1. The table shows that there is only one combination for which the function F_1 is 1 otherwise it is equal to 0.

Truth Table 6.11 (a) $F_1 = A B \overline{C}$, (b) $F_2 = A + \overline{B} C$, (c) $F_3 = A B C + A \overline{B} C$, (d) $F_4 = \overline{A}\,\overline{B} C + \overline{A} B C + A \overline{B}$, and (e) $F_5 = A \overline{B} + \overline{A} C$.

ROW	A	B	C	F_1	F_2	F_3	F_4	F_5
# 1	0	0	0	0	0	0	0	0
# 2	0	0	1	0	1	0	1	1
# 3	0	1	0	0	0	0	0	0
# 4	0	1	1	0	0	0	1	1
# 5	1	0	0	0	1	0	1	1
# 6	1	0	1	0	1	1	1	1
# 7	1	1	0	1	1	0	0	0
# 8	1	1	1	0	1	1	0	0

Let us consider another expression (b) representing F_2.

$$F_2 = A + \overline{B} \cdot C$$

Here the value of F_2 will be 1 if either A is equal to 1 or if $\overline{B} \cdot C$ will be 1 when $B = 0$ and $C = 1$. Table 6.11 shows that $A = 1$ in the last four rows and B. Therefore there are five such combinations for which $F_2 = 1$.

Consider another function F_3 given by the expression:-

$$F_3 = A B C + A \overline{B} C$$

The function F_3 will be equal to 1 when $A = 1$, and $C = 1$ as shown in the 6^{th} row when B is equal to 0 and also in the 8^{th} row when B is equal to 1.

Let us consider function F_4 which is given by the expression:-

$$F_4 = \overline{A}\,\overline{B} C + \overline{A} B C + A \overline{B}.$$

The function F_4 will be equal to 1 when $A = 0$, $B = 0$, and $C = 1$ for which the first term $\overline{A}\,\overline{B} C$ is equal to 1 as shown in the 2^{nd} row or F_4 will be equal to 1 when $A = 0$, $B = 1$ and $C = 1$ for which the second term $\overline{A} B C$ is equal to 1 as

shown in the 4^{th} row. F_4 will also be equal to 1 when A = 1, and B = 0 for the third term $A\overline{B}$ is equal to 1 as shown in the 5^{th} and 6^{th} rows. Thus the four combinations of the values for variables that make $F_4 = 1$ as shown in the 5^{th} and 6^{th} rows.

Consider the function F_5 which is given by the expression:-

$$F_5 = A\overline{B} + \overline{A}C.$$

The function F_5 will be equal to 1 when A = 1, B = 0, for which the term $A\overline{B}$ is equal to 1 as shown in the 5^{th} and 6^{th} rows or F_5 will be equal when A = 0, and C = 1 for which the second term $\overline{A}C$ will be equal to shown in the 2^{nd} and 4^{th} rows. Therefore there are four combinations for the function $F_5 = 1$. These combinations are the same as that for function F_4.

6.10 SIMPLIFICATION OF BOOLEAN FUNCTION

This Boolean expression is simplified as much as possible using Boolean rules. For example, consider the functions F_4 and F_5

$$F_4 = \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}$$

And $$F_5 = A\overline{B} + \overline{A}C$$

The Truth Table 6.10 shows that F_5 is the same as F_4. Both have identical results in each case for all the possible combinations of 0's and 1's of the binary variables A, B and C. **Two functions of n binary variables are said to be equal if they have the same result for all the possible combination of variables.** Thus function F_4 can be implemented by using the Boolean expression for F_5 which give the same result as that of F_4 for the same number of variables. The expression for F_5 is simpler than that for F_4 since it contains less number of **literals**. A **literal** is a primed or unprimed variable in an expression.

6.11 ALGEBRAIC MANIPULATION

Another way to simply a Boolean expression is factoring the expression by using Boolean rules. For example consider function F_3.

$$F_3 = A\,B\,C + A\overline{B}\cdot C$$

Factoring the above expression for simplification.

$$\begin{aligned} F_3 &= A\cdot B\cdot C + A\cdot\overline{B}\cdot C \\ &= A\cdot(B+\overline{B})\cdot C \\ &= A\cdot 1\cdot C && \text{[by Boolean axiom 5(a)]} \\ &= A\cdot C && \text{[by Boolean axiom 1(b)]} \end{aligned}$$

Which is the simplified expression for F_5

Consider function $F_5 = \bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B}$

$= \bar{A} \cdot (B + \bar{B}) \cdot C + A \cdot \bar{B}$

$= \bar{A} \cdot 1 \cdot C + A \cdot \bar{B}$ [by axiom 5(a)]

$= \bar{A} \cdot C + A \cdot \bar{B}$ [by axiom 1(b)]

or $F_5 = A \cdot \bar{B} + \bar{A} \cdot C$ [by axiom 2(a)]

EXAMPLE 6.2

Simplify the Boolean function F given below to minimum number of literals.

$F = A + \bar{A} B$

SOLUTION: $F = A + \bar{A} \cdot B$

$= (A + \bar{A}) . (A + B)$ [by axiom 4(b)]

$= 1 \cdot (A + B)$ [by axiom 5(a)]

$= (A + B) \cdot 1$ [by axiom 2(b)]

$= A + B$ [by axiom 1(b)]

EXAMPLE 6.3

Simplify the Boolean function F given below to minimum number of literals.

$F = A \cdot \bar{B} + \bar{A} B C + \bar{A} \cdot \bar{B} \cdot C$

SOLUTION $F = A \cdot \bar{B} + \bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C$

$= A \cdot \bar{B} + \bar{A} \cdot (B + \bar{B}) \cdot C$ [by axiom 4(a)]

$= A \cdot \bar{B} + \bar{A} \cdot 1 \cdot C$ [by axiom 5(a)]

$= A \cdot \bar{B} + \bar{A} \cdot C$ [by axiom 1(b)]

EXAMPLE 6.4

Simplify the Boolean function F given below to minimum number of literals.

$F = A \cdot B + \bar{A} \cdot C + B \cdot C$

SOLUTION: $F = A \cdot B + \bar{A} \cdot C + B \cdot C$

$= A \cdot B + \bar{A} \cdot C + B \cdot C$ [by axiom 1(a)]

$= A \cdot B + \bar{A} \cdot C + B \cdot C\,(A + \bar{A})$ [by axiom 5(a)]

$= A B + \bar{A} C + B C A + B C \bar{A}$ [by axiom 4(a)]

$= A B + A B C + \bar{A} C + \bar{A} B C$ [by axiom 2(a),(b)]

$= A B (1+C) + \bar{A} C (1+B)$ [by axiom 4(a)]

$= A . B . 1 + \bar{A} . C . 1$ [by theorem 2(a)]

$= A . B + \bar{A} . C$ [by axiom 1(b)]

6.12 MINIMIZATION OF BOOLEAN FUNCTION BY MAPPING

When a Boolean function is implemented by logic gates, it is always desirable to use minimum number of gates. It not only reduces cost but also increase speed and reliability of the circuit. We have seen how Boolean algebra can be applied in getting simpler Boolean expression. If the truth table is available, the expression can also be simplified by the mapping technique suggested by Maurice Karnaugh.

KARNAUGH MAPS

Karnaugh maps are also used to simplify Boolean expressions. karnaugh maps transform a truth table into a simplified expression. Karnaugh mapping method helps to minimize the number of literal in a Boolean expression and thus reduces the number of gates required in digital circuit for its implementation.

Let us consider functions F_1 and F_2 with two variables A and B such that:-

$F_1 = AB + A\bar{B}$ and $F_2 = A B + \bar{A}\ \bar{B}$

$F_1 = 1$ if $A = 1$ and $B = 1$ or $A = 1$ and $B = 0$ while

$F_2 = 1$ if $A = 1$ and $B = 0$ or $A = 0$ and $B = 0$

The above functions F_1 and F_2 are given in the truth table 6.12. In this Truth Table we have four possible combinations for the variables A and B.

Truth Table 6.12 for $F_1 = AB + A\bar{B}$ and $F_2 = A\bar{B} + \bar{A}\,\bar{A}$

COMBINATION	A	B	F_1	F_2
# 1	0	0	0	1
# 2	0	1	0	0
# 3	1	0	1	1
# 4	1	1	1	0

Let us construct a map of $2 \times 2 = 4$ squares consisting of two columns and two rows. Each square will be allocated one of the four combinations of the variables A and B such that the variables A, B take the value 1 and their complementary $\bar{A}$ and $\bar{B}$ take the value 0 as shown below:-

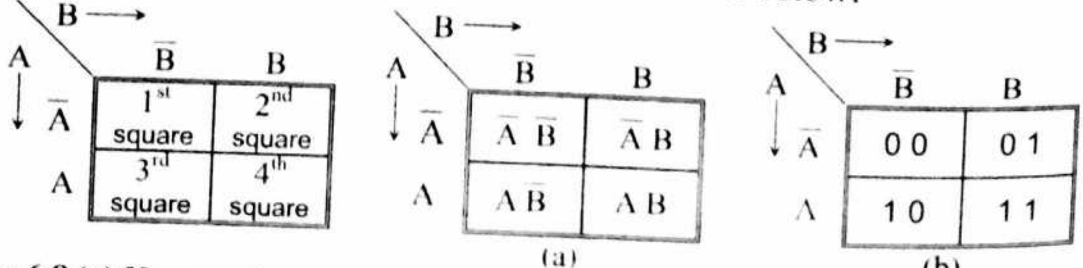


Figure 6.8 (a) Karnaugh map in term of variable, (b) in term of the value of the variables.

In the karnaugh map top left square represents the values 1st square of the truth table top right square represent the 2nd square, lower left square represent the 3rd square and lower right square represents the 4th square. The terms present in the Boolean expression for $F_1 = 1$ are those assigned to 3rd square and 4th square.

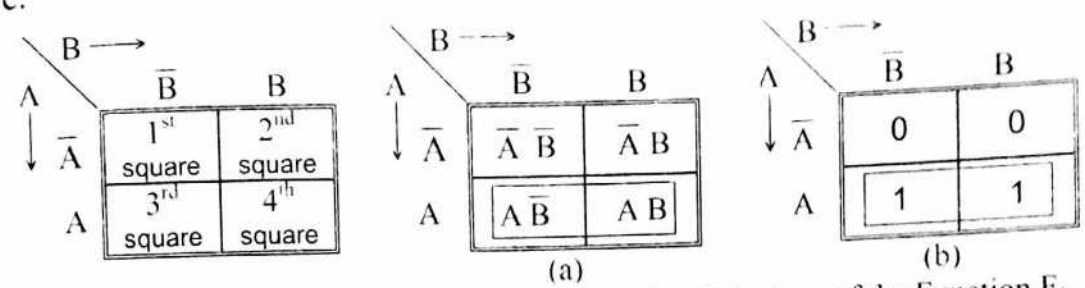


Figure 6.9 (a) Karnaugh map in term of variable, (b) in term of the Function F_1

The square No. 3 and No. 4 is a group having value = 1 for the function F_1, independent of the value of the variable B i.e. whether B = 0 or B = 1. The value of the function F_1 is independent of the variable B so the variable B can be dropped from the expression. Thus

$$F_1 = AB + A\overline{B}$$

or $$F_1 = A$$

Similarly the terms present in the Boolean expression for F_2 are those in squares 1 and 3 for which the value of $F_2 = 1$

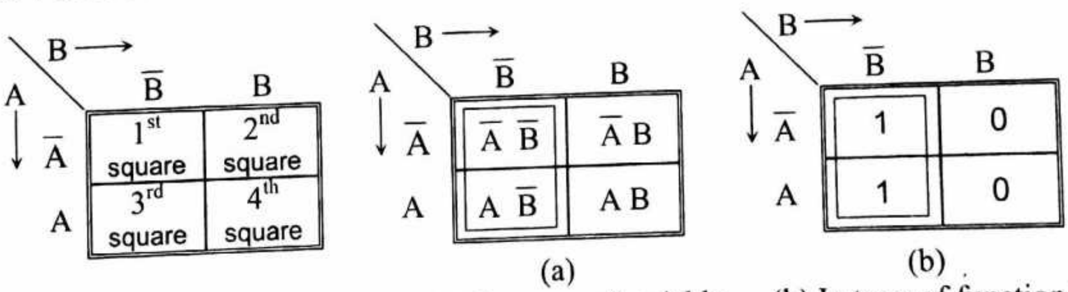


Figure 6.10 Karnaugh map (a) In terms of variable (b) In term of function F_2

The group of square 1 and 3 is a group whose value for the function $F_2 = 1$ is independent of the variable A i.e. it does not matter whether A = 0 or A = 1. Thus dropping variable A and $\overline{A}$ from the expression of F_2 we get simplified expression for F_2 as

$$F_2 = A\overline{B} + \overline{A}\,\overline{B}$$
$$= \overline{B}$$

EXAMPLE: 6.5. Consider a logic function F with three variables A, B and C such that $F = ABC + A\overline{B}C + \overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$

SOLUTION:

Truth Table 6.13 for $F = ABC + A\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}C$

ROW	A	B	C	F
# 1	0	0	0	0
# 2	0	0	1	1
# 3	0	1	0	0
# 4	0	1	1	1
# 5	1	0	0	1
# 6	1	0	1	1
# 7	1	1	0	0
# 8	1	1	1	1

Karnaugh map consisting of 8 squares is constructed to represent eight possible combination of the three variables as shown in the truth table 6.13.

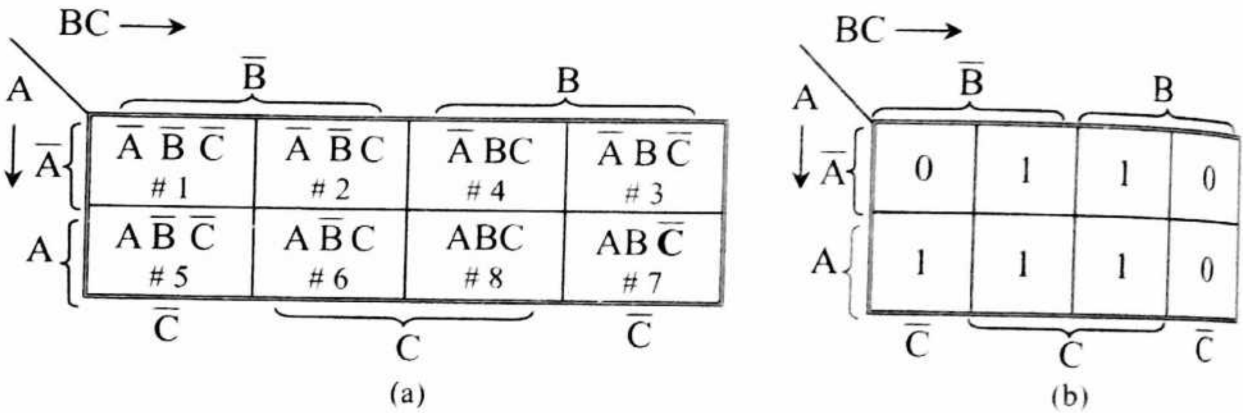


Figure 6.11 Karnaugh map (a) in terms of variable (b) in terms of the function **F**

The eight squares of the map are representing these combinations such that in the first row all the four squares take the literal $\bar{A}$ and all the squares in the second row take A. In the first two columns all the four squares have $\bar{B}$ and all the four squares in the 3rd and 4th columns have B similarly the four squares in the 1st and 4th square have literal $\bar{C}$ while the four squares of column 2nd and 3rd have literal C as shown in Figure 6.11(a). All the terms present in the expression are assigned value 1. Thus square # 2, 4, 5, 6 and 8 are assigned value 1 as shown in Figure 6.11 (b) corresponding to row # 2, # 4, # 5, # 6 and # 8 in the truth table 6.4 above. Thus the function is represented by Karnaugh map with five squares having value 1 and remaining three squares having value 0.

As a rule the variable, which differs in a group of two adjacent squares in a karnaugh map, is dropped. In case of a group of four adjacent squares two variables which differ in that group are dropped.

There are two groups of squares in Figure 6.11 (b). One group of squares # 2, # 4, # 6, and # 8 in the middle and the second group of squares # 5 and # 6 in the lower left end. The first group has four adjacent squares enclosing four 1's in the middle of the K-map and second group has two adjacent squares enclosing two 1's on the lower left corner of K-map. In the first case the group of four adjacent squares in a karnaugh map differs in two variables A and B. Hence both these variables A and B are dropped and we are left with C only. In the case of second group of two adjacent squares # 5 and # 6, F differs in C only and hence C is dropped leaving behind $A\overline{B}$. Thus F will be given by:-

$$F = ABC(8) + A\overline{B}C(6) + \overline{A}BC(4) + A\overline{B}\,\overline{C}(5) + \overline{A}\,\overline{B}C(2)$$
$$= \{\overline{A}\,\overline{B}C(2) + \overline{A}BC(4) + A\overline{B}C(6) + ABC(8)\} + \{A\overline{B}\,\overline{C}(5) + A\overline{B}C(6)\}$$

After dropping the variables that differs within the groups, we get

$$F = C \text{ (in the First group)} + A\overline{B} \text{ (in the Second group)}$$
$$= A\overline{B} + C$$

Groups consisting of 2 squares in a map can be formed as shown in Figure 6.12(a) and groups consisting of 4 squares in a map can be formed similar to the one shown in Figure 6.12(b) and (c)

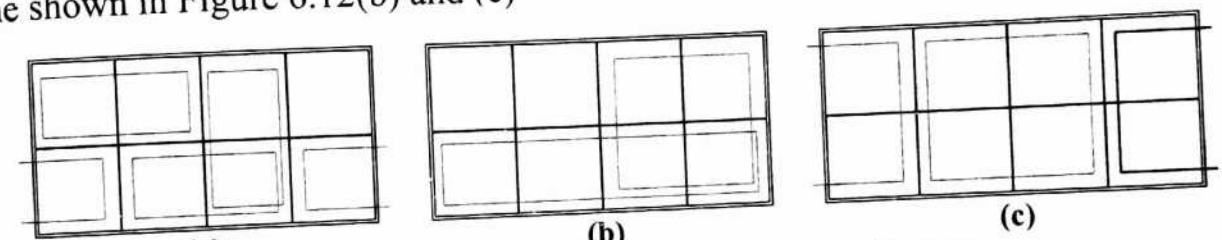

(a) **(b)** **(c)**

Figure 6.12 Karnaugh maps showing grouping of square

The following steps may be followed for mapping and simplification of the Boolean Function:-

1. Identify the square corresponding to each term in the Boolean expression.
2. Enter value 1 in the respective square of the karnaugh map.
3. Identify adjacent 1's to Form group.
4. Each two adjacent 1's may be combined in a group to eliminate one literal.
5. Each four adjacent 1's may be combined in a group to eliminate two literals.
6. Final expression consists of logical addition of the terms representing the groups in the karnaugh map.

EXAMPLE 6.6. Simplify the Boolean function F using karnaugh mapping

$$F = ABC + AB\overline{C} + A\overline{B}\,\overline{C} + \overline{A}BC$$

SOLUTION:

The karnaugh maps for function F is shown in Figure 6.13(a), (b) and (c).

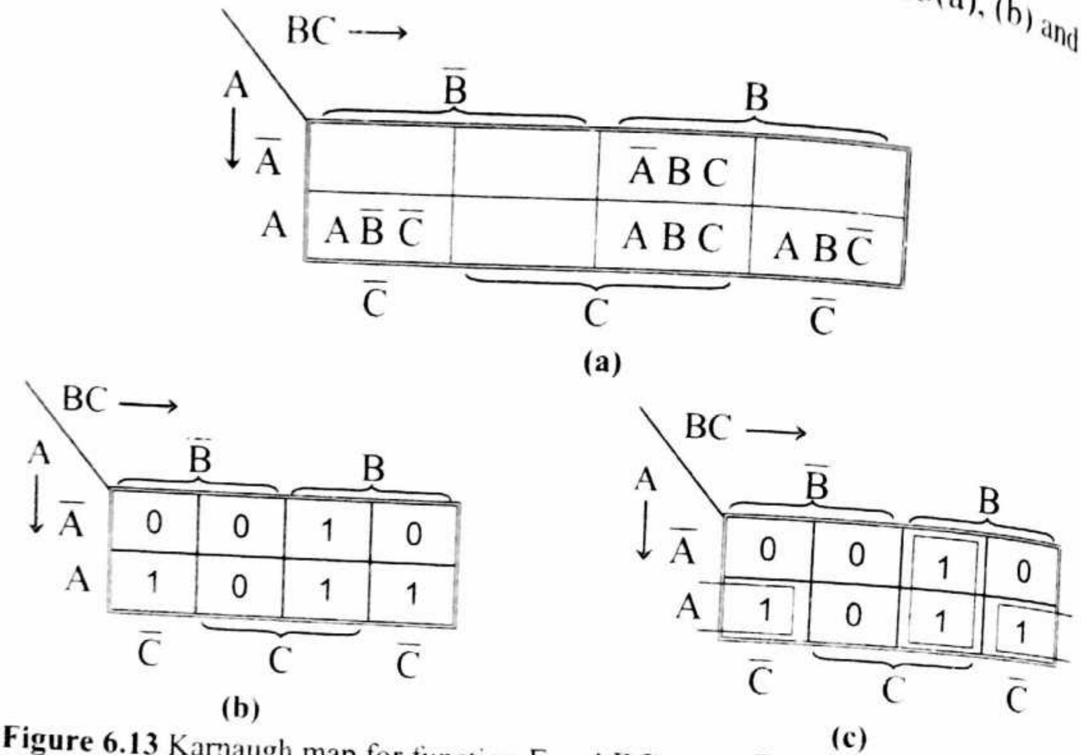


Figure 6.13 Karnaugh map for function $F = ABC + AB\overline{C} + A\overline{B}\,\overline{C} + \overline{A}BC$

The K-map for the function F consists of 8 squares. Four squares are marked 1's for each term present in the function F. While the remaining four squares are marked 0's.

Two adjacent 1's of the lower left square (square #5) and lower right square (square #7) are grouped to eliminate literal B leaving behind $A\overline{C}$.

Two adjacent 1's in column #3 (square #4 and square #8) are combined in another group to eliminate one literal A leaving behind BC.

Since $F = ABC(8) + AB\overline{C}(7) + A\overline{B}\,\overline{C}(5) + \overline{A}BC(4)$

$F = (A\overline{B}\,\overline{C}(5) + AB\overline{C}(7) + (ABC(8) + \overline{A}BC(4)$

$= [A.\overline{C}(\overline{B} + B)] + [(A + \overline{A}).BC]$

Hence $F = A\overline{C} + BC$

EXAMPLE 6.7.

Simplify the Boolean function F. using Karnaugh Mapping.

$$F = ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + \overline{A}BC + \overline{A}\,\overline{B}C$$

SOLUTION:

Three variable Karnaugh map for function F is shown in Figure 6.14.

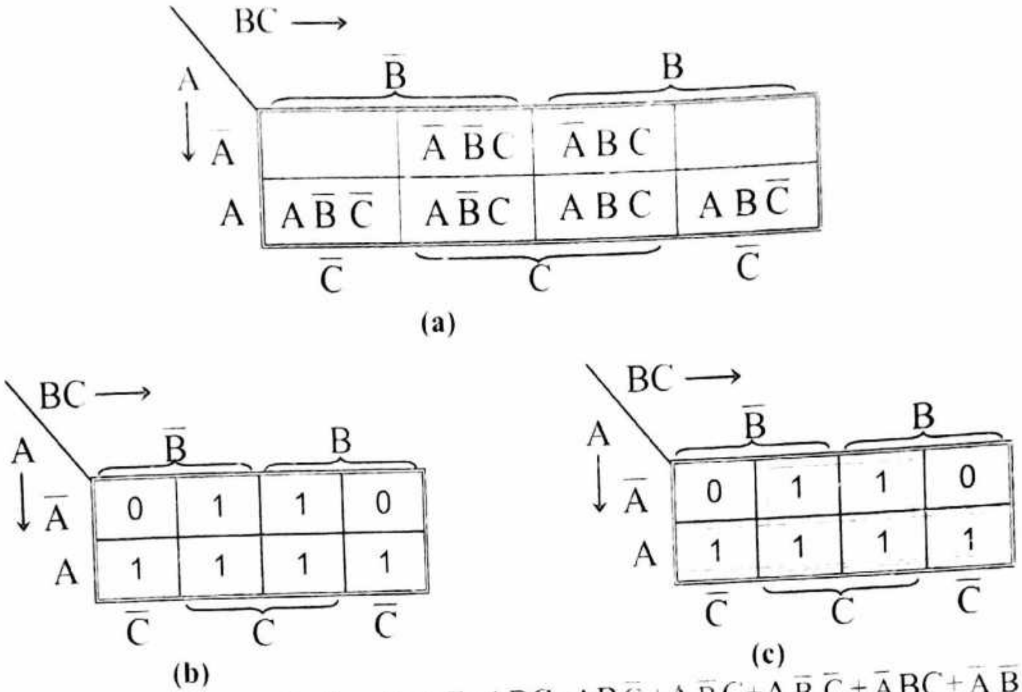


Figure 6.14 Karnaugh map for function $F=ABC+AB\bar{C}+A\bar{B}C+A\bar{B}\bar{C}+\bar{A}BC+\bar{A}\bar{B}C$

There are six squares marked with 1's for each term of the function F for which F = 1. For remaining input combinations in the K-map, F = 0.

There are four adjacent 1's in the second row and are grouped together. This eliminates two literals B and C leaving behind only the literal A. Similarly there are four adjacent 1's in the second and third column and are also grouped together to eliminate two literals A and B leaving with literal C. Thus we get literal A from the first group and C from the second group. Hence

$$F = A + C$$

SUMMARY

George Boole, presented it as mathematical logic known as Boolean Algebra. The electronic switching devices / circuits have two discrete **ON** and **OFF** states. The **two-valued Boolean algebra** is also called **Switching Algebra**.

ELEMENTS OF BOOLEAN ALGEBRA

In Boolean algebra an expression can be formed using binary variables, constants, symbols of logic operations and parentheses.

BOOLEAN VARIABLES

The variables used in Boolean algebra can be represented by letters of the alphabet having values; **1** or **0**

BOOLEAN CONSTANTS

In Boolean algebra a set of constants has only two elements **0** or **1**. The Boolean constant is either 0 if not 1 or is 1 if not 0.

LOGICAL OPERATORS

In Boolean algebra there are three basic operators **NOT**, **AND** and **OR**.

NOT OPERATORS: **NOT** operation is a unary operation which is the **negation** of the value of a variable on which NOT operates

OR OPERATION: In Boolean algebra **OR** operation represents **logical addition**

AND OPERATION: In Boolean Algebra **AND** operation represents **logical multiplication**.

BOOLEAN EXPRESSION

An expression is a logical statement which is either true or false. These statements are represented by variables operated by logical operators.

OPERATOR'S PRECEDENCE

A Boolean expression can be evaluated using the following precedence:-

i) Expressions must be scanned from left to right.
ii) Parentheses are evaluated first.
iii) **NOT** operations are performed after parentheses.
iv) **AND** operations are performed next to **NOT**.
v) **OR** operations are performed at the end.

IDENTITY ELEMENTS

The element **0** is an **identity element** with respect to binary operation **OR**. The element **1** is an **identity element** with respect to binary operation **AND**

LAWS OF BOOLEAN ALGEBRA

These laws consist of postulates and the theorems of Boolean Algebra. The postulates are the basic axioms of the Boolean Algebra and need no proof.

POSTULATES OF BOOLEAN ALGEBRA

Some of the basic axioms of Boolean algebra are:-

Axiom 1. a) $A + 0 = A$ and b) $A \cdot 1 = A$

Axiom 2. a) $A + B = B + A$ and b) $A \cdot B = B \cdot A$

Axiom 3. a) $A + (B + C) = (A + B) + C$ and b) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

Axiom 4. a) $A \cdot (B + C) = A \cdot B + A \cdot C$ and b) $A + (B \cdot C) = (A + B) \cdot (A + C)$

Axiom 5. a) $A + \overline{A} = 1$ and b) $A \cdot \overline{A} = 0$

Duality Principle The axioms of Boolean algebra are in pairs. One part may be obtained from the other if the binary operators (+ and .) as well as the identity elements (0 and 1) are interchanged. This property is called the duality principle.

TRUTH TABLE

A truth table is a table that shows the result of a Boolean expression for all the possible combinations of the values given to the variables used.

THEOREMS OF BOOLEAN ALGEBRA

The theorems of Boolean algebra used to simplify Boolean expressions are:-

THEOREM 1. (a) $A + A = A$ (b) $A \cdot A = A$

THEOREM 2. (a) $A + 1 = 1$ (b) $A \cdot 0 = 0$

THEOREM 3. (a) $A + (A \cdot B) = A$ (b) $A . (A + B) = A$

THEOREM 4. $A = \overline{\overline{A}}$

THEOREM 5. a) $\overline{A + B} = \overline{A} \cdot \overline{B}$ (b) $\overline{A \cdot B} = \overline{A} + \overline{B}$

BOOLEAN FUNCTIONS

A Boolean function is an expression which can be formed by binary variables, the binary operators OR, AND, NOT, parentheses and equal sign. The function assumes the value either 0 or 1 for given values of the variables.

SIMPLIFICATION OF BOOLEAN FUNCTION

The basic approach is to proceed from a statement of the function to a truth table and then to a Boolean expression of the functions. This Boolean expression is then simplified as much as possible using Boolean rules.

Two functions of n binary variables are said to be equal if they have the same result for all the possible combination of n variables.

ALGEBRAIC MANIPULATION

Another way to simply a Boolean expression is factoring the expression by using Boolean rules.

MINIMIZATION BY KARNAUGH MAPPING

Karnaugh mapping method helps in minimizing the number of literals in a Boolean expression. The following steps may be followed for mapping and simplification of the Boolean Function:-

1. Identify the square corresponding to each term in the Boolean expression.
1. Enter value 1 in the respective square of the karnaugh map.
2. Identify adjacent 1's to Form group.
3. Each two adjacent 1's may be combined in a group to eliminate one literal.
4. Each four adjacent 1's may be combined in a group to eliminate two literals.
5. Final expression consists of logical addition of the terms representing the groups in the karnaugh map.

As a rule the variable, which differs in a group of two adjacent squares in a karnaugh map, is dropped. In case of a group of four adjacent squares two variables which differ in that group are dropped.

6.04 Match the items given in Column I with those given in Column II

	Column I		Column II
i)	AND	a)	1
ii)	OR	b)	$A + B$
iii)	$A + A$	c)	Union
iv)	$A + AB$	d)	A
v)	$A + (A \cdot B)$	e)	Intersection

6.05 What is Boolean algebra.

6.06 What do you understand by the logical operations AND and OR.

6.07 Find the values of the Boolean expressions

i) $XY + X\overline{Y}$ when $X = 1$ and $Y = 0$

ii) $(X + Y).(XY)$ when $X = 1$ and $Y = 0$

iii) $(X + \overline{Y}).(\overline{X} + Y)$ when $X = 1$; $Y = 1$

6.08 State and prove the two basic Demorgan's theorem. Find out the complement of the following Boolean expressions:

i) $XY + X\overline{Y}$ (ii) $(X + Y)\cdot(XY)$ (iii) $(X + \overline{Y})\cdot(\overline{X} + Y)$

6.09 What is a Truth Table? Construct a truth table for AND and NOT of AND operation for the three variables X, Y and Z.

6.10 State and prove the following laws:

i) Idempotent Law (ii) Involution Law (iii) Absorption Law

6.11 Construct a truth table for the following Boolean expression.

i) $XY + \overline{X}Z + YZ$ (ii) $(X + Y).(XY)$ (iii) $X\overline{Y} + XZ + Y\overline{Z}$

6.12 Simplify the following Boolean expression.

i) $\overline{A}C + \overline{A}.B + A\overline{B}C + BC$ (ii) $X\overline{Y}Z + X\overline{Y}\overline{Z} + \overline{X}\overline{Y}Z + \overline{X}\overline{Y}\overline{Z}$

iii) $(A + B + C)\cdot(A + \overline{B} + \overline{C})\cdot(A + B + \overline{C})\cdot(A + \overline{B} + C)$

6.13 Simplify the following function using karnaugh map.

i) $A\overline{B}C + ABC + \overline{B}\,\overline{C}$ (ii) $ABC + A\overline{B}C + AB\overline{C}$

ii) $A\overline{B}C + \overline{A}B\overline{C} + \overline{A}\,\overline{B}C$ (iii) $A\overline{B} + \overline{A}C + B\overline{C}$

EXERCISES

6.01 Complete the following statements

i) In Boolean algebra plus sign stands for __________ operation.

ii) AND operation is used for logical __________.

iii) Double complementation has __________ effect.

iv) In Boolean Algebra 1+1+1 is equal to__________.

v) According to the distributive law { A + (B · C)} = __________.

6.02 Tick (✓) the following statements either True or False.

i) AND operation is also called as logical addition True/False

ii) In a NOT gate, the output is negative of the input True/False

iii) Two series switches can represent AND operation. True/False

iv) **NOT** of (A + B) is equal to **NOT** of A **OR** **NOT** of B True/False

v) A variable that is same within two adjacent squares of a Karnaugh map is dropped. True/False

6.03 Encircle one choice A, B, C or D in each case.

i) Boolean algebra is also known as

(A) Logical algebra (B) Code Algebra
(C) Switching algebra (D) Digital Algebra

ii) An OR operation has 4 variables. The possible number of combinations in its truth table are:-

(A) 4 (B) 8 (C) 16 (D) 32

iii) The output will be one if all inputs except one are zero in case of:-

(A) NOT operation (B) AND operation
(C) OR operation (D) NOT of OR

iv) According to the Demorgon's law $\overline{A + B}$ is equal to:-

(A) $\overline{A} + \overline{B}$ (B) A . B (C) $\overline{A} . \overline{B}$ (D) A + B

v) According to absorption law A · (A + B) is equal to:-

(A) 1 + B (B) A B (C) A (D) A + 1

COMPUTER SOFTWARE

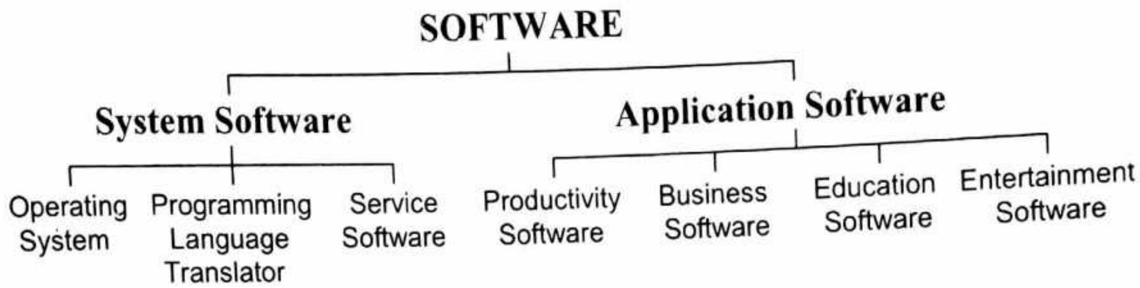
Computer is one of the most versatile machine that has enhanced the mental capability of humans and has relieved him much from mental labour. A computer can produce typeset documents, manipulate data, design spacecrafts, control and regulate industrial plants, diagnose diseases, forecast weather, control, provide world-wide data communication, and much more.

What a computer can do depends largely on its software. Software can be referred to as the driving force of computer without which it would be of little value to the user.

Computer hardware is useless without software. Similarly software is useless without hardware. It is necessary that hardware and software must work together to obtain desire results.

7.1 TYPES OF SOFTWARE

Software can be classified into two main groups; **System Software** and **Application Software.** These are further divided into various categories as shown in the chart below.



7.2 SYSTEM SOFTWARE

System software includes the programs which control the operation of a Computer system. They direct the fundamental operations of the Computer system like; displaying information on the screen, storing data on disks, sending data to the printer, interpreting the typed commands and communicate with peripheral devices. In addition System software provides a foundation so that application software operates correctly and use peripheral devices properly. System software can be divided into three main groups.

- **Operating System**
- **Programming language Translator**
- **Service Software**

OPERATING SYSTEM

An operating system is a collection of System programs that controls and coordinates the overall working of a computer system. It manages the resources of a computer system such as the Central Processing unit, Memory and Input/Output devices. An operating system acts as an interface between the Hardware and the Applications software of a computer. Examples are Disc operating system (DOS), Macintosh, and Unix.

PROGRAMMING LANGUAGE TRANSLATOR

It has been discussed earlier that computers can understand machine language only which is in 0 s and 1 s format. Early programs were written in

machine languages. Machine languages are very difficult and differ from computer to computer. Next to Machine languages, assembly languages and high-level languages were developed.

A language translator is a System software that converts assembly language/high-level programs called **source programs** into a low-level machine language called **object programs**. During the past fifty year nearly five hundred computer languages have been developed. Every programming language has its own translator that is different from others. A language translator may be an Interpreter or a Compiler as already discussed in Chapter I.

SERVICE SOFTWARE

Service programs make the life of a user a bit easier. They help to manage hardware, software, and data resources in a better way. Service software includes Utility programs, Library programs and Device drivers

UTILITY PROGRAMS

Although an operating system usually works behind the scenes, there are times when you need utilities or utility software, to do things such as prepare disc to store data, detect computer viruses on a disc, prepare, copy, merge, sort out and update files. Protects monitor screen from burning during idle monitors by screen saver utility program. This prolongs the life of the monitor. A text editor utility program such as WordStar, WordPerfect, Norton's editor etc., is used to create and modify a document Another popular utility program called data compression, (e.g. WinZip) helps us reduce the size of a stored image/text for more efficient transmission over a data communication link.

LIBRARY PROGRAMS

Library program maintains a directory of frequently used software module and their locations. These programs might consist of manufacturer-supplied or user-written routines or complete programs to compute mathematical functions such as root, exponential function and matrix inversion, control input/output devices, maintain appointments and so on. The library programs make these routines available when requested by the user, or by the operating system/ application program.

DEVICE DRIVERS

Device drivers are the programs, which are required to run peripheral devices such as mouse, keyboard, floppy disk, CD-ROM modem, printer etc. these software tell your computer how to use your input/output devices.

7.3 APPLICATION SOFTWARE

Application software also referred as productivity software is developed to help the user to increase his/her productivity. The **application packages** were designed for the kinds of data management tasks that most organizations need for their business-related activities such as record keeping, inventory control, and payroll accounting. There has been a tremendous increase in the use of application software packages with the development of microcomputers. Application software can be divided into the following four major classes.

- **Productivity software**
- **Business software**
- **Education software**
- **Entertainment software**

PRODUCTIVITY SOFTWARE

The most important type of the application software is productivity software. It helps you to work most effectively. Productivity software includes: -

- Word Processing Software
- Spreadsheet Software
- Database Management Software
- Graphics Software
- Communications Software and
- Integrated Software

BUSINESS SOFTWARE

A second major application software category, business software, helps organizations efficiently accomplish routine tasks. Accounting and payroll applications are good examples of business software.

EDUCATION AND REFERENCE SOFTWARE

Education and reference software is designed to help you learn more about a particular topic. For example it may help you how to type, how to use word processor, or how to speak French. Educational simulations let you work with a computerized model of something in the real world. Reference software, such as electronic encyclopedias can also help you to look up facts on any topic.

ENTERTAINMENT SOFTWARE

Some application packages are designed for entertainment. With these entertainment softwares, one can play cards, fly a simulated jet, play 18 holes of golf, solve a Sherlock Holmes mystery, battle monsters, or explore new worlds.

7.4 OPERATING SYSTEM

Operating System provides link between the user and the computer. It enables the user to have complete control over the computer system by commands through the keyboard or any other input device. The supervisor part of an

operating system resides all the times in the memory (RAM) of a computer after the computer is turned on. An operating system performs the following tasks:-

- *Controls basic Input and Output devices*
- *Allocates System Resources*
- *Manages Storage Space*
- *Supports Application software*
- *Maintains Security*
- *Detects Equipment failure.*

This enables the user to achieve high efficiencies of which the processing systems are capable. Operating systems can be classified as:

- Single User operating systems.
- Multi-user operating systems.

SINGLE USER OPERATING SYSTEMS

Single user operating systems are used on individual systems or Personal computer. These operating systems reside on disks and are called Disk Operating Systems (DOS). Some of the most widely used single user operating systems are:-

- MS-DOS
- PC DOS
- Macintosh System
- DOS Shell
- OS/2
- Windows

MULTI-USER OPERATING SYSTEMS

Multi-user operating systems are used for data processing on large computer systems. They enhance the potential of computer systems with multiprogramming, multiprocessing and timesharing facilities making them suitable for large organizations. Some of the multi-users operating systems are:-

- UNIX/XENIX
- LAN
- WAN
- WINDOWS NT

7.5 DISK OPERATING SYSTEM (DOS)

The single user operating system most popular in microcomputers upto mid of 1990's is **disk operating system (DOS)**. It is called DOS because it is stored on the disk of a computer. The major tasks the DOS must carry out are:-

- *to control input and output devices*
- *to enable user to load and execute program, and*
- *to maintain an orderly system of data on the disk.*

COMPUTER FILES

Computers store data in the form of files. Computer files are of two types:-

Data Files: *Data files contain the documents and graphics that can be viewed, saved, printed, edited, copied or sent. These are generally created by the use of application software.*

Program Files: *Program files contain the instructions that direct a computer for a specific task. These include operating system, utilities, and application software. These are the executable files because the computer performs or executes the instructions in these files.*

DOS FILES

One of the primary role an operating system is to manage files. DOS has three special files that you may see when working on your PC. These three types of files are designated by their extension, that is BAT, COM, or EXE, and are called **Batch files**, **Command files** and **Executable files** respectively. These are executable files and contain DOS commands/programs.

DOS COMMANDS

DOS consists of a series of program designed to control the computer's resources. DOS commands can have three parts: Filename, parameters and switches. Mostly these commands consist of regular English like words. For example ERASE, COPY, DEL, DIR, PRINT, etc. There are two types of DOS commands. These are DOS Internal commands and DOS External commands.

DOS INTERNAL COMMANDS

DOS internal commands are stored in the COMMAND.COM file. These are loaded automatically into the memory (RAM) during boot up process when you start your system. These include the simpler and common commands that are needed on regular basis. Internal commands are part of COMMAND.COM. Thus you never see their names in a directory listing. These commands remain resident in the RAM and are available to you all the time. Some of the internal DOS commands are: CLS, DIR, DEL, DATE, TIME, EXIT etc. You will find labs on DOS Internal commands in the Computer Lab. Journal for SSC. Computer Science by Prof. M. Tahir Hassan written in a simple and easy to follow way.

DOS EXTERNAL COMMANDS

The external DOS commands are those, which need a special file for its execution. Mostly the DOS commands that are normally not used frequently except for special operations are given as external commands. The three types of files that can run as an external command are COM, EXE and BAT files. Some of

the external DOS commands are: CHKDSK, DELTREE, FORMAT, XCOPY etc. You will find labs on DOS External commands in the Computer Lab. Journal for SSC. by the same author written in a manner that is simple and easy to follow.

PARAMETERS

Sometimes DOS requires additional information, which is specified in one or more parameters after the command name. Parameters indicate items that the command should operate on. For example, the **del** (erase) command requires a parameter, which names the file you want to delete. To delete a file named NOTES.TXT. You would type del notes.txt.

SWITCHES

Switches modify the way a command performs. Switches are usually entered using a switch specifier followed by a forward slash (/). A switch specifier usually consists of a single letter or number. For example suppose you want to view a listing of a directory that contains large number of files. When you enter **dir** command, the filenames scroll on the screen so rapidly that you find it difficult to read. If you add the switch [/**P**] a list of files will appear one screen at a time. Using the switch [/**W**] a list of filenames will appear on a wide format.

DOS INTERNAL COMMANDS

CLS Clears the screen

Syntax CLS

CHDIR (CD) Displays the name of the current directory or changes the current (default) directory

Syntax CHDIR [*drive:*] [*path*]

cd [*drive:*][*path*]

Parameters [*drive:*][*path*] Specifies the drive (if other than the current drive) and directory to which you want to change.

COPY Copies one or more files to another location.

Syntax COPY [*file name*] [*source:*] [*destination* [*directory*]]

DATE Displays the date or allows you to change the date form your terminal or from a batch program.

Syntax DATE [*mm-dd-yy*] When you enter the date command, the current date is displayed and you are requested to enter a new date.

DEL (ERASE) Deletes one or more specified files from the disk.

Syntax DEL [*drive:*][*path*] *filename* [/**p**]

ERASE [*drive:*][*path*] *filename* [/**p**]

Parameters [*drive:*][*path*] Specifies the location and name of the file or set of files you want to delete.

Switch /**p** Prompts you for confirmation before deleting the specified file. When you use the /**p** switch, **DEL** displays the name of a file and prompt to you with a message: *filename*, delete (Y/N)?

DIR Displays a list of current directory files and subdirectories.

Syntax DIR [*drive:*] [*path*] [*filename*][/**p**][/**w**]

Parameters [*drive:*][*path*] Specifies the drive and directory for which you want to see a listing.

[*filename*] specifies a particular file or group of files for which you want to see a listing.

Switch /**p** Displays one screen of the listing at a time. To see the next screen, press any key.

/**w** Display the listing in a wide format.

EXIT Exits the command processor (COMMAND.COM Program) and returns to a previous program that started COMMAND.COM, if it exists.

Syntax EXIT

REM (BATCH COMMAND) Enables you to include comments (remarks in a batch file or in your CONFIG.SYS file

Syntax REM

TIME Displays the system time or sets your computers internal clock.

Syntax: Time [*hours*:[*minutes*:[*seconds*[.*hundredth*]]] [a|p]]

Parameters *hours* specifies the hour (from 0 to 23).

minutes specifies minutes (from 0 to 59).

Second specifies seconds (from 0 to 59).

Hundredth specifies hundredths of a second (from 0 to 99).

a|p a specifies A.M or P.M for the 12 hours time format.

TYPE Displays the contents of a text

Syntax TYPE [*drive*:] [*path*] *filename*.

Parameter[*drive*:] [*path*] *filename*.

Specifies the location and name of the file that you want to view.

VER Displays the DOS version number

Syntax VER

VOL Displays the volume label and serial number if they exist.

Syntax VOL [*drive*:]

Parameter *drive*: specifies the drive that contain the disk for which you want to display the label and serial number.

DOS External Commands

BACKUP Backs up one or more files from a fixed disk to a floppy or new disk.

Syntax BACKUP *source*: [*destination- drive* :][/s][/m][/F]

CHKDSK Checks the validity of a disk and reports the total space, number of files and amount of free space on the disk and the total and available amount of RAM in the computer.

Syntax CHKDSK [*drive*:] [*path*] [/f] [/v]

COMMAND Starts the command processor.

Syntax COMMAND [[*drive*:]*path*][*device*][/e:*nnnnn*][/p] [/c *string*] [/msg]

COMP Compare the content of two sets of files.

Syntax COMP [*file-spec 1*] [*file- spec 2*] [/d] [/a] [/l] [/n]

DELTREE Deletes all files and subdirectories in a directory.

Syntax DELTREE [/Y][*pathname*]

(If you do not include the /Y switch, Deltree will prompt you to type Y to deletes all directories or N to leave the directories intact.

DISKCOPY Duplicates the content of one Disk to another.

Syntax DISKCOPY [*source drive*:] [*target drive*:] [/l] [/v]

EDIT Starts DOS Editor, which creates and changes ASCII text files.

Syntax EDIT [[*drive*: [*path*] *filename*] [/b][/g][/h][/nohi]

FC Compare two files and displays the differences between them.

Syntax FC [/a /b] [/c] [/ [*drive* 1:] [*path 1*] [*filename 1*] [*drive 2*:] [*path 2*] [*filename 2*]

FORMAT Formats the disk in the specified drive to accept DOS files

Syntax FORMAT [*drive*:][/v: *label*][/q][/u][f: *size*][/t: *tracks* /n: *sectors*] [/b][/s]

GRAPHICS Initializes DOS for printing graphics.

Syntax GRAPHICS [*type*] [[*drive*:][*path*] *filename*] [/r] [/b] [/lcd] [/printbox:std|/printbox:lcd]

type Specifies the type of printer.

[*drive*:][*path*] *filename* Specifies the location and name of the printer profile that contains information about all supported printers.

PRINT Prints text files using a Queue.

Syntax PRINT [/d:*device*][/b:*size*] [/u:*ticks1*] [/m:*ticks2*] [/s:*ticks3*] [/q:*qsize*] [/t] [[*drive*:][*path*]*filename*[…]] [/c] [/p]

PROMPT The PROMPT command defines how the DOS command prompt appears onscreen.

Syntax PROMPT [*prompt_string*] (characters with a specific value such as)

$t Current time
$d Current date
$p Current drive and path
$n Current drive
$_ ENTER-LINEFEED
$e ASCII escape code (code 27)

$n Current drive

RESTORE Restores Backed up files from Floppy to Hard Disk.

Syntax RESTORE *drive1*:*drive2*:[*path*[*filrname*]] [/s] [/p] [/b:*date*] [/a:*date*] [/e:*time*] [/l:*time*] [/m] [[/n] [/d]

SYS Creates a Bootable drive and installs the main DOS files.

Syntax SYS [*drive1*:] [*path*] *drive2*:

TREE Displays directory structure: Graphically displays the path of each directory and sub directory on the drive.

Syntax TREE [*drive*:] [*path*] [/f] [/a]

XCOPY Copies files(except hidden and system files) and directories, including subdirectories.

Syntax XCOPY *source* [*destination*] [/a|/m] [/d: *date*] [/p] [/s] [/e]] [/v] [/w]

CREATE OR MAKE DIRECTORY

To create a directory use the MD or MKDIR (make directory) command. This command enables you to build the subdirectory structures used for organizing and managing files. The following command creates a new directory named DRAWING under the current directory.

MD DRAWING

You can use either MD or its longer form, MKDIR, with the same result. The MD command makes a sub-directory in the current directory, unless you specify otherwise. As with CD (change directory) command you may use a drive specifier and a directory path specifier to create sub-directory under other than current directory, backslash(\) for root directory and two periods(.. for a parent directory can be used.

For example: If the current directory is HOME. To make a directory called \HOME\DRAWING you can type the following.

MD DRAWING

CHANGING DIRECTORIES

To move to a different directory on the current drive, use CD or CHDIR (Change directory) command. For example the following command changes the current directory to \SCHOOL\WORK\ART:

CD\SCHOOL\WORK\ART

CD command to change to the parent directory as follows:

CD:

DISK OPERATING SYSTEM (DOS)

The most popular Single user operating systems upto mid of 1990's were **disk operating system**. It is called DOS as it is stored on the disk of a computer.

COMPUTER FILES

A Computer store data in the form of files. These files are of two types; Data files and the Program files.

DOS COMMANDS

DOS consists of a series of programs that control the computer's resources. There are two types of DOS commands.

DOS INTERNAL COMMANDS

DOS internal commands are loaded automatically into the memory (RAM) during boot up process when you start your system. These include CLS, DIR, DEL, DATE, TIME, EXIT etc. that are needed on regular basis.

DOS EXTERNAL COMMANDS

Usually external DOS commands are required less frequently. These reside on disk and are loaded into RAM only when needed. Some of the external DOS commands are: CHKDSK, DELTREE, FORMAT, XCOPY etc.

CREATE OR MAKE DIRECTORY

To create a directory use MD or MKDIR (Make directory) command. This command enables to **create** or **make** a new **directory.**

CHANGING DIRECTORIES

To move to a different directory on the current drive, use CD or CHDIR (Change directory) command.

DELETING DIRECTORIES

To **delete** a directory, use RD or RMDIR (Remove directory) command. The directory to be removed must be empty and should not be the current directory.

PATH

The path specifies the location of a file within the directory tree. You can think of it as the route DOS must follow, starting at the root directory, to get to files in another directory. The path command allows you to generate a list of directories that DOS searches to locate programs.

EXERCISES

7.01 Complete the following statements

i) An _________ acts as an interface between the hardware and the applications software

ii) Device drivers is a _______________ software.

iii) DOS internal commands are stored in the _________ file.

iv) Type _______ after the prompt to change the default drive to drive A.

v) CHDIR is a DOS _________ command.

vi) CHKDSK is a DOS _________ command.

7.02 Tick (✓) the following statements either True or False.

i) Compiler is a type of application software True/False

ii) A program in high level language is called an object program. True/False

iii) Database management software is an application Software True/False

iv) Operating system resides in RAM when the Computer is ON. True/False

v) DOS shell provides a GUI. True/False

vi) The default can never be changed. True/False

7.03 Encircle one choice A, B, C or D in each case.

i) Interpreter is a type of
(A) System Software (B) Service Software
(C) Utility Software (D) Application Software

ii) Which of the following is the characteristic(s) of an operation system?
(A) Controls Basic input output devices
(B) Allocate system resources
(C) Managing storage space
(D) All of the above

iii) Which of the following is not a single user operating system?
(A) MS DOS (B) PC DOS (C) DOS Shell (D) None of the above

iv) Which of the following is not an executable command?
(A) CLS (B) REM (C) REN (D) COPY

v) Which of the following is a DOS internal command?
   (A) ERASE (B) EDIT
   (C) DELTREE (D) SYS

7.04 Match the items given in Column I with those in Column II

	Column I		Column II
i)	Loading DOS	a)	Format Disk
ii)	Switch	b)	Compare file
iii)	TYPE	c)	Find a file
iv)	FORMAT	d)	Booting
v)	FC	e)	Modifies Command
		f)	Print the current file
		g)	Contents of text

7.05 Clearly define the terms, data and information, give an example in each case not mentioned in the book.

7.06 Write the names of various types of data, give examples in each case.

7.07 What is meant by machine language? Explain.

7.08 In what respect a high-level language is different from a low-level language.

7.09 Name at least four high-level languages and describe any one of these high-level languages.

7.10 What is the significance of an assembly language?

7.11 What is an operating system? Mention various types of operating systems.

7.12 What is DOS?

7.13 What is the difference between DOS and DOS Shell?

7.14 Name the types of commands used in DOS and describe at least five commands of each type.

7.15 What is meant by booting the system?

8

INTRODUCTION TO WINDOWS

Every computer needs an operating system that controls its basic **input and output** operation, such as receiving commands from the input devices and displaying information at the output devices. It enables the user to communicate with the computer system. Microsoft Windows operating system enables you to operate your computer with ease. It has gone gradual up gradation such as Windows 95, Windows 97, Windows 98, and Windows 2000.

Windows is a **graphical user interface** (GUI) and is easy to learn in contrast to the DOS, which is **command line Interface**. An **interface** determines how you interact with your computer to do things like entering data or running programs like changing fonts, numbering lines, breaking the text into columns or spell checking. In addition to Windows, several other operating systems are also available for microcomputers, including MS-DOS, OS/2, Windows NT, Unix and system 7 for the Macintosh computers.

Once the operating system is loaded, it takes **charge** until you shut down the computer. An operating system performs the following functions: -

- *Provides the instructions to display the on screen elements with which you interact.*
- *Loads programs such as word processing, spread sheet programs into the computer's memory for your use.*
- *Coordinates how programs work with central processing Unit, keyboard, mouse, printer and other hard ware as well as with other software.*
- *Manages the way the information is stored on and retrieved from disks.*

8.1 THE DESKTOP

The Desktop appears when you load Windows as shown in Figure 8.1. This contains small pictures or graphic objects and a bar at the bottom. The bar has a start button and other things on it. The screen is called desktop because it looks like a desk with items you might have on your own personal desk.

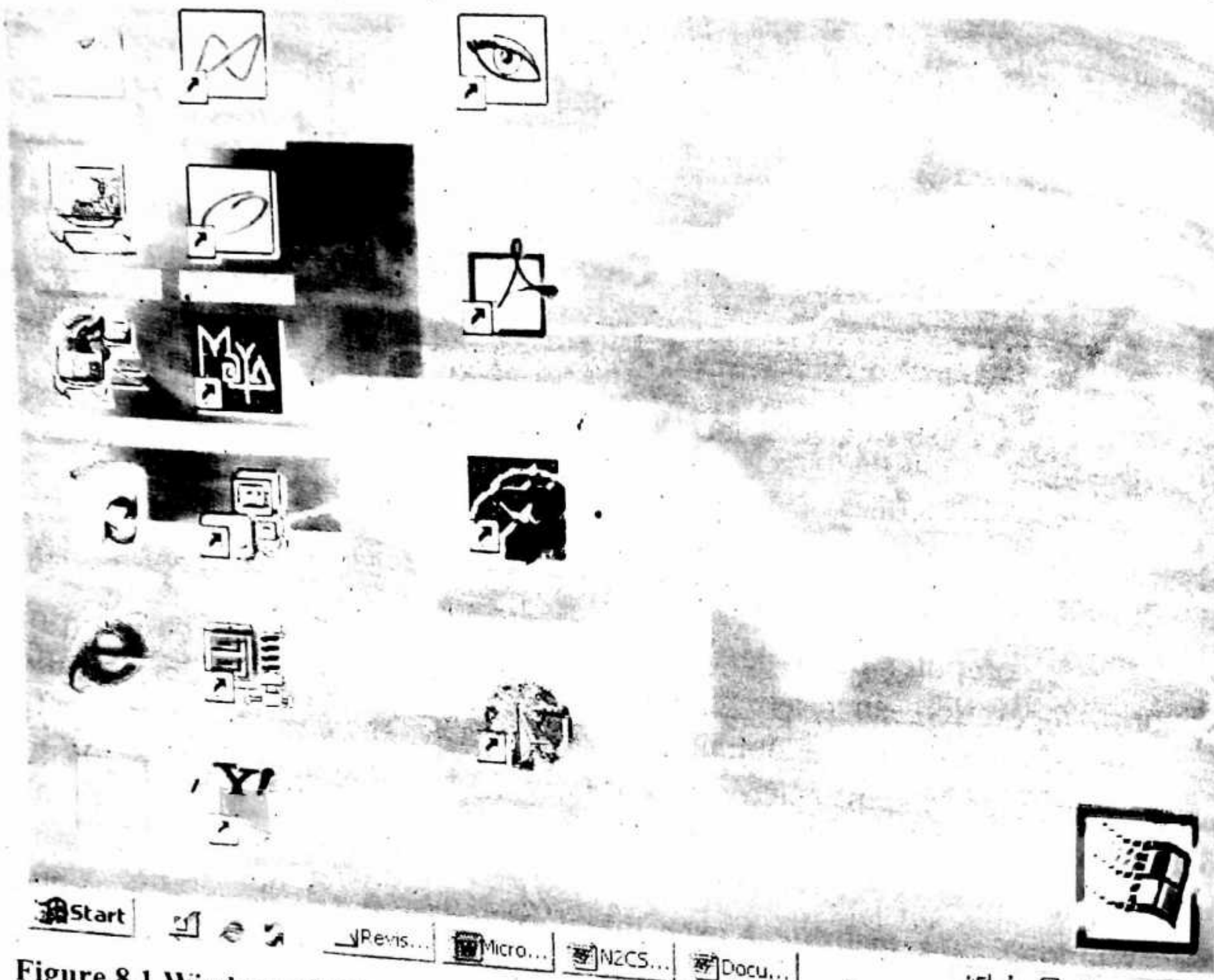


Figure 8.1 Windows 2000 Initial screen (Desktop) with background wallpaper, Icons and Task bar. (You may have slightly different items depending on options you choose when installing Windows)

ICONS

You see small graphic objects such as My Computer, a Recycle Bin, My Briefcase on the Desktop. These objects are called Icons, meaning images. Icons represent the parts of the computer system when you work using a GUI. An icon is used to carry out specific task on the desktop with the help of a keyboard or a mouse. We can divide these icons into various groups such as:-

- System icons
- File icons
- Application icons
- Drive icons
- Folder icons
- Command icons

BUTTONS

There is another class of symbols that is used in a GUI called buttons. Buttons is an area of screen where clicking causes something to happen. Most buttons have names while some buttons have icon or both name and icon surrounded by a black border. Start button  has both name and icon.

8.2 HOW TO USE THE KEYBOARD ?

Keyboard is an input device for entering text. It also offers shortcut methods for performing some common commands and procedures. For example the cursor movement keys let you change the position of the cursor and select items on the screen. The function keys (F1, F2, ... F12) allow you to give the computer commands without typing long strings of characters.

8.3 HOW A MOUSE IS USED?

An arrow [ ] appears on the screen when mouse is connected with your system. This is the mouse pointer. As you move the mouse over a flat surface, the pointer moves in the windows. Push the mouse forward, the pointer moves up. Move the mouse to the left, the pointer goes to the left. The pointer moves in direction in which you move the mouse. On the top of the mouse are two or three buttons that are used to make selections. The most common mouse actions are:-

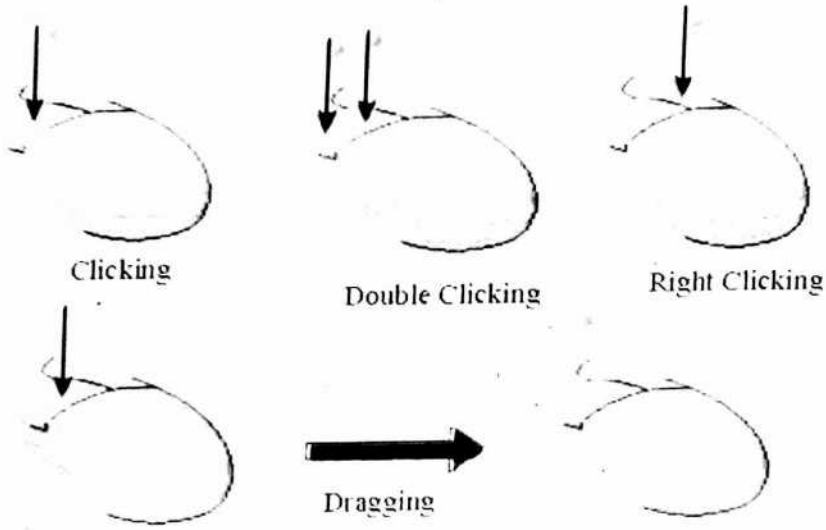


Figure 8.2 Various mouse actions

- **Pointing** **Move the mouse on your desk to position the tip of the mouse pointer over the desired object on the screen.**
- **Clicking** **Press and release the mouse button without moving the mouse.** Clicking may be used to select icons, to position the cursor, to select menu commands and to choose an option in a menu dialog box.
- **Double-clicking** **Press and release the mouse button twice in rapid succession without moving the mouse.** Double-clicking in Windows may be used to activate a drive or to select a file/folder. It may also be used to select and activate an object such as My computer and Recycle Bin on the desktop.
- **Right-clicking** **Press and release the right mouse button without moving the mouse.** Right clicking on an icon or word may display a shortcut menu.

- **Dragging** **Press and hold the mouse button while you move the mouse pointer across the screen**. When the pointer reaches the desired location, release the mouse button. Dragging is used to move objects or Windows and to create shortcuts for objects.

You will notice that sometimes mouse pointer changes its shape on the screen as you move it at different parts of the screen or when an application performs a certain task. Each shape of the mouse pointer has its own purpose. A mouse pointer shape may provide some important information. The most common mouse pointer shapes are:-

Left Arrow	It is used to select objects, choose menu commands, access buttons on the taskbar and application toolbars, and dialog boxes.
Hourglass	It informs you that windows is busy in performing an assigned task and requests you to wait.
I-beam	It is used to modify and edit text and to position the insertion point. It is also called a cursor.
Hand	It is used to select shortcuts and definitions in the Help window.
Move	It is used to drag the table to the new location

While working in Windows you may also find other mouse pointer shapes. In Windows the characteristics of mouse pointers changes with its shape.

8.4 LOADING WINDOWS

This session assumes that you are working on computers that use Windows. When you turn on your computer, Windows is loaded automatically into the computer's memory (RAM) from the hard disk where it resides. This process of loading the operating system is called booting the system. After a few seconds, the initial Windows desktop appears. A welcome dialog box appears with usability tips as shown in Figure 8.3 each time you load Windows, unless the feature has been disabled. Click the close button at the top right hand corner on the window title bar to get started the Windows session.

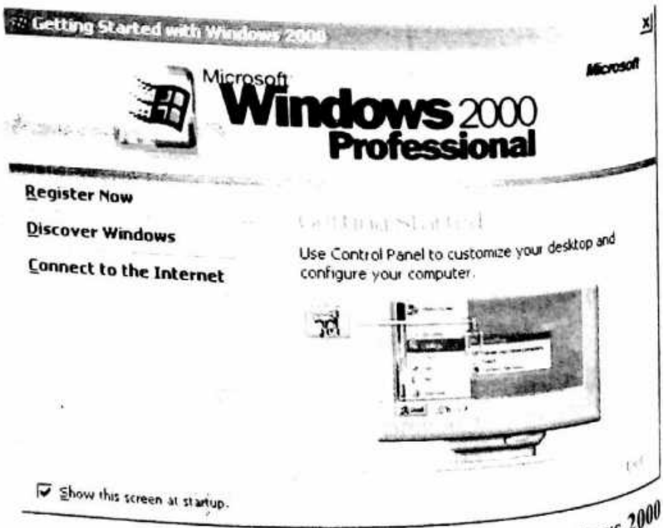


Figure 8.3. A Welcome dialog box of Windows 2000

8.5 FAMILIARIZATION WITH WINDOWS ICONS: (Desktop)

Windows desktop includes number of icons on the left and a horizontal bar normally positioned at the bottom of the screen. This bar is known as **taskbar**. The icons on the desktop allow you to view and interact with windows. Some of the icons that may appear on the desktop in Windows are:-

 My documents

 My computer

 Network Neighbourhood

 Internet Explorer

 Recycle Bin

 My Briefcase

 Netscape Communicator

 Window Explorer

 The Control panel

 Virus Scan

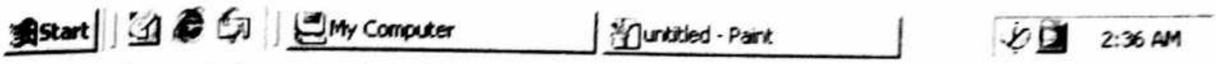
 Willow TALK 2.5

 Win Zip

 Connect to the Internet

 Yahoo Messenger

THE TASK BAR AND THE START BUTTON



The task bar can be placed at any of the four sides of your screen or can be made to disappear. It reappears when you move the pointer to the far edge of the screen or press **Ctrl-Esc.** The Task bar has two primary functions:-

- It acts as a launching pad for all the application being open.
- It keeps in track all the working application being performed.

As you open more programs, the taskbar automatically resizes its icon view to ensure that all the opened programs can always be seen. You can shift from one program to the other by clicking on the desired program button that is being displayed on the task bar.

THE START BUTTON 

The start button is a permanent feature of the task bar located at its left end. This button activates the main menu of Windows known as **Start Menu**. This includes almost everything that is installed on your computer from

Figure 8.4 A pull up Start menu

system tools and application software to shut down option as shown in Figure 8.4. A tool with right pointing arrow-head ▸ that appears next to the menu command indicates that a cascading menu will be displayed on selection.

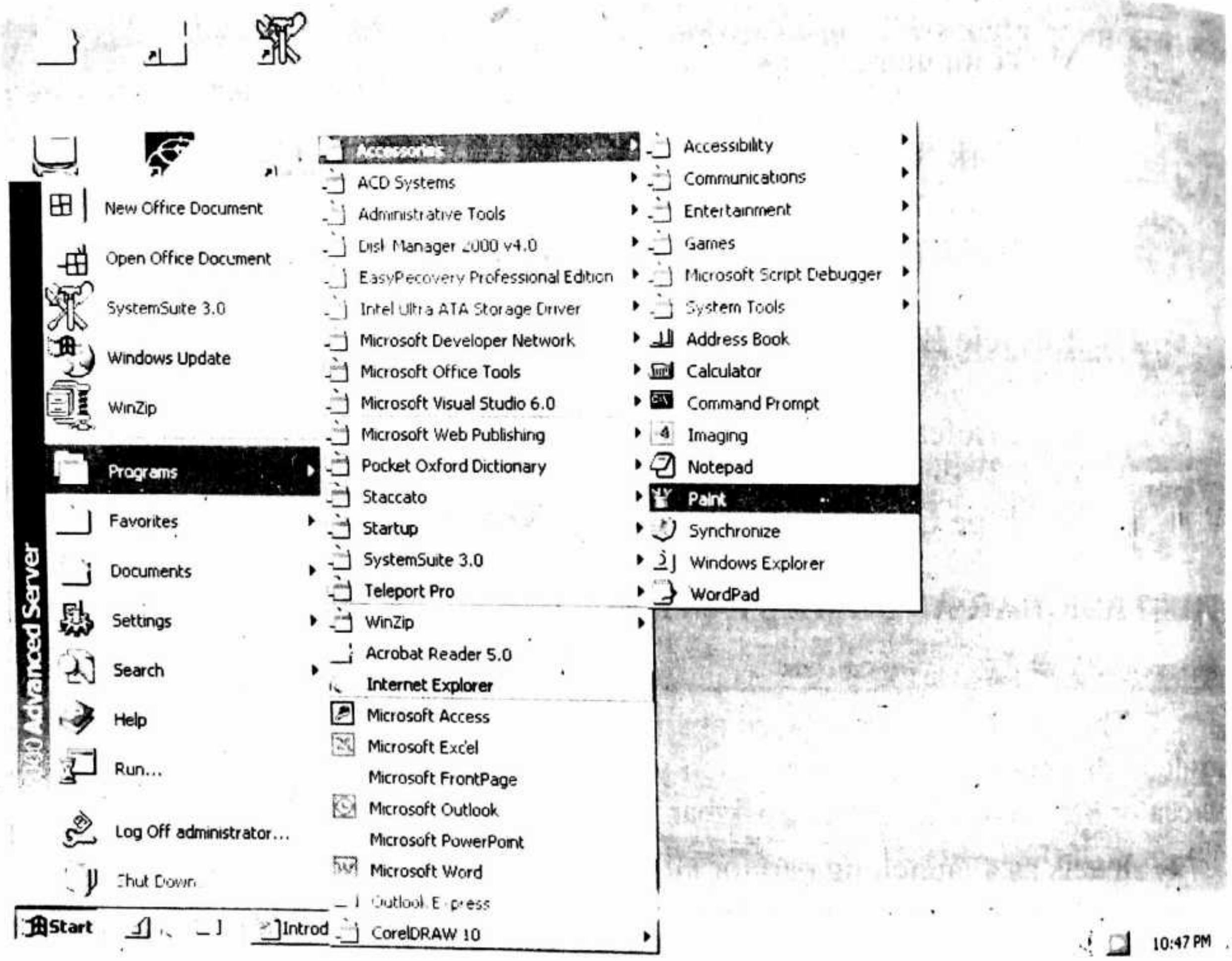


Figure 8.5 showing **start menu** with various **cascading** and **sub-cascading menus.**

Some of these options may have **sub-cascading menu** as shown in Figure 8.5 where **Paint** program is being highlighted. For example, point the mouse pointer to the **Programs** command. A **cascading menu** will appear to the right of the start Menu at the **Programs** command. With the mouse pointer on the **Programs** command, **slide the pointer** to the right until a **highlight** appears in the **Program cascading menu.** Highlighting an item such as **Accessories** from **Program cascading menu** will cause a **sub-cascading menu** to appear.

8.6 COMPONENTS OF A WINDOW

This section introduces you with some of the common characteristics or **components** of a window. For Labs. on Windows consult **Computer Lab. Journal** for SSC. Computer Science by Prof. M. Tahir Hassan.

TITLE BAR

The title bar, located at the top of a window contains the name of the application. The name and icon of the **Control Panel** appears at the left on the title bar as shown above. When you have opened many windows, the title bar also differentiates an **active window** from an **inactive window**. The title bar of an active window has a darker colour than the title bar of an inactive window.

CONTROL BUTTONS

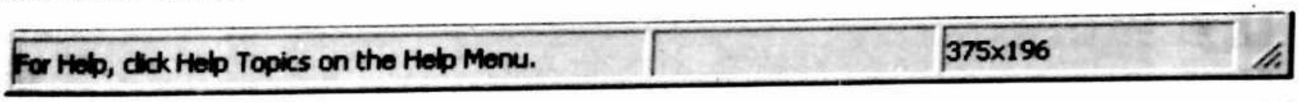
Every window has a set of three control buttons that appear in a row on the right side of the title bar. These are the **Minimize, Maximize/Restore and close buttons** . If you point at one of these buttons, a **control menu** appears telling you the action that can be taken on clicking it.

You can minimize a window from the view by clicking the **Minimize** button when currently it is not required but must remain running.

You can **Maximize** window by clicking the maximize button that enlarges the window to fill the entire screen. When a Window is maximized, a **Restore** button appears in its place. With the **Restore** button you can return the Window to its original size.

A window has an associated button having its name with an icon on the taskbar. You can close a window by clicking the **Close** button .

STATUS BAR



The **status** bar that appears at the bottom of a window, provides useful information about the window. For example, the **Paint**'s status bar tells you how the function of various commands where the mouse pointer has last been moved and the pixels position of the cursor on the canvas of the **paint's window**.

SCROLL BARS

When there is more information that cannot fit into a window at a single time, **scroll** bars are displayed at the right or bottom borders. You can use **scroll bars** to move around in a window by clicking the **arrow-heads** , , , at the ends of a scroll bar or by dragging **scroll box** located on the **scroll** bar.

MENU BAR

The **Menu** bar appears immediately below the **title** bar (some windows do not have Menu bar). **Commands** are grouped together on the **Menu** bar for manipulating information in the window. The **Menu** bar is accessed by clicking on a desired command using the mouse. When **Menu** bar is activated by choosing any command, a **drop-down menu** appears below it that lists all its commands. Any command that is not available for selection on the **drop-down menu** appears **dim** (usually in light grey colour). A check mark besides a command means that it is currently active or that the feature is enabled.

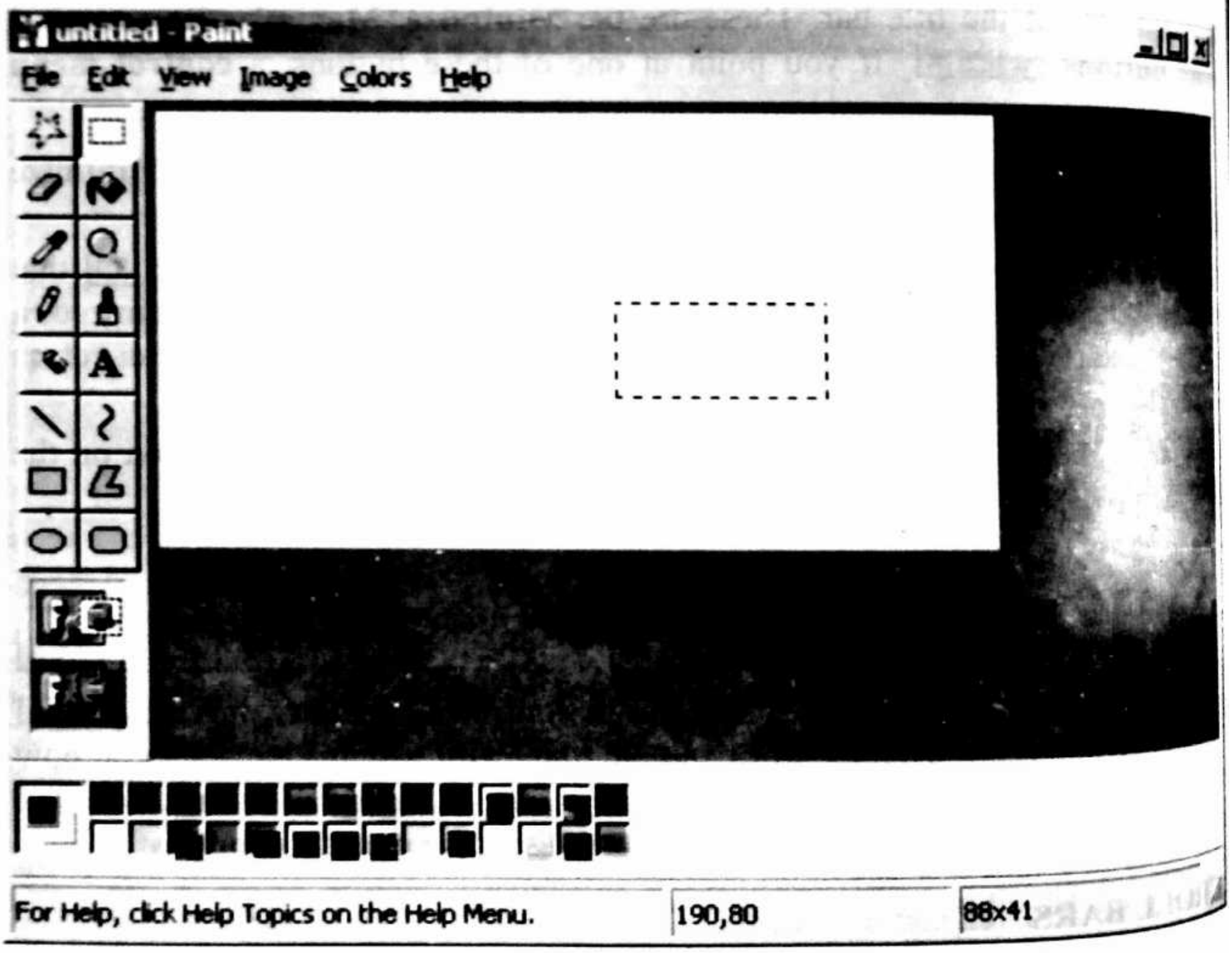


Figure 8.5 showing Paint Windows with various tool bars in

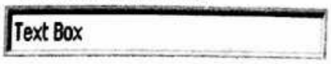
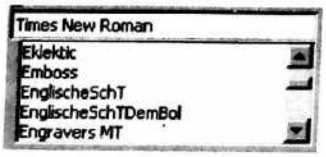
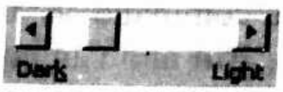
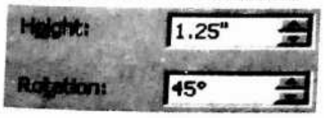
DIALOG BOXES

A dialog box is a common element in a Window application that collects information before processing a command or instruction. Dialog boxes appear on the screen when a running application needs more information before continuing.

A dialog box uses several methods for collecting information as described in Table 8.1 Some dialog boxes ask you to enter information (such as file names) while others simply require you to check off options or make choices from a list. After you enter the request and click OK, windows start executing the command.

Dialog box is also used to display message or ask to confirm a command. In a dialog box, you enter text and select options. Dialog boxes are sometimes composed of multiple tabs that allow you to access additional pages within the dialog box by simply clicking on the named tab. Tab pages keep a dialog box to a reasonable size while still letting you adjust a lot of settings from it. Click the **OK** command button to proceed when you have finished making selections.

Table 8.1. Showing various kinds of dialog boxes

NAME	EXAMPLE	ACTION
Text box		Type information into the box
Check box		Click the box to turn it on or off. A check mark appears in the box when the option is turned on.
Option buttons		Select an exclusive option form a group of options
Command buttons		Command buttons are used to execute a command immediately after they are selected. Such as OK, Apply, Cancel etc.
List Box		Make a choice from the scroll-able list box. Multiple choices are visible at the same time. Only the currently selected option is highlighted.
Drop-down list box		Select an option from a drop-down list by clicking the down arrow next to the textbox; only the currently selected option is visible.
Slide box		Drag the slider bar to make a selection. The text below the slider bar often changes to guide you as you move along the bar.
Spin box		Click the spinner controls or up and down arrows untill text box displays the required number.

8.7 MY COMPUTER 

My Computer is an important addition to the interface. My computer shown in Figure 8.6 enables you to see easily what is in your computer. Every Windows Screen has an icon by this name at the top left corner. Double click on it and you can access to every thing in your computer hardware as well as software.

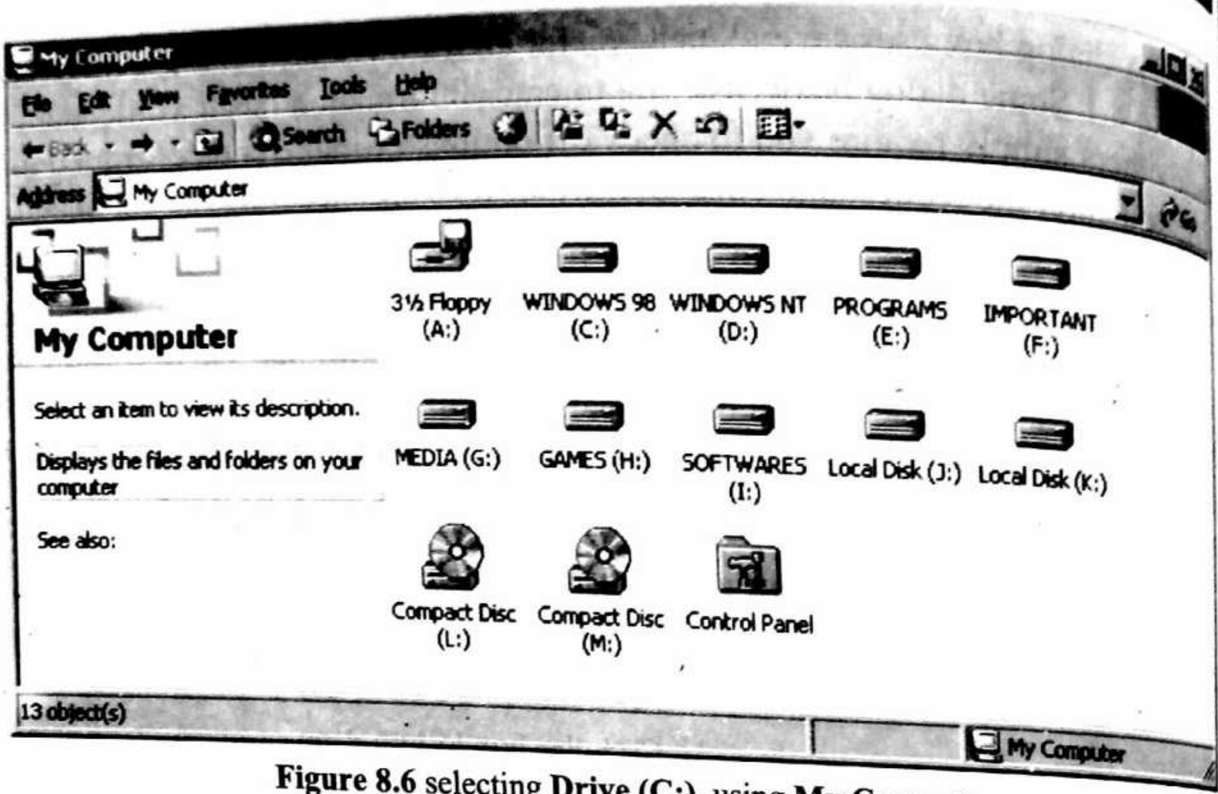


Figure 8.6 selecting **Drive (C:)** using **My Computer**

My Computer allows access to computer resources from hardware devices to file folders. Such as **Floppy drive** (A), **Disk Drive** (C), **Disk Drive** (D), **Compact Disk** (E), **Control Panel** etc. All these services can also be accessed through other means but using **My Computer**, it is easy to discover and manage these applications.

8.8 THE RECYCLE BIN 

When you **right-click** an object, you may see a **Delete** command in the menu. This command is different than the cut command as Delete sends the selected files, folders or other objects to **Recycle Bin** and Cut command puts the file on the clipboard for pasting to some other location.

When you delete a file, folder, or some other item, you click on it once and press the **Delete** key or drag it to the **Recycle bin** icon on the desktop. If you prefer to use menu, you can **right-click** files icon and then choose the Delete Command from its shortcut menu. Before actually removing your file, window asks you to confirm through a dialog box.

Any file you delete from the hard disk goes straight to the Recycle bin. The Recycle bin temporarily holds the items that you delete until you either empty the Recycle bin or remove it from the Recycle bin to use it again if you have accidentally deleted it. Click Recycle Bin and find the item you want to restore. Click the Restore command to undelete it.

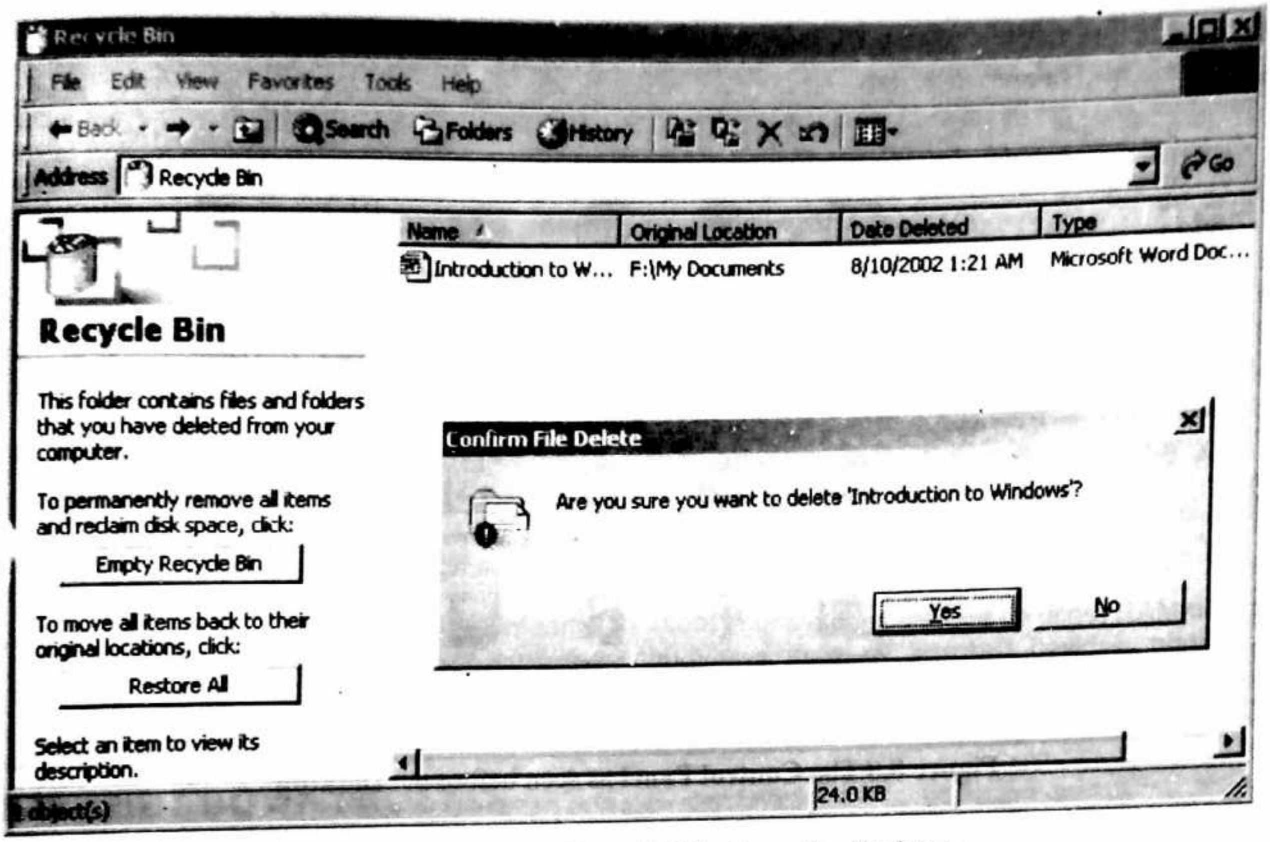


Figure: 8.7 selecting Recycle Bin from the desktop.

You can delete the file permanently by selecting it in the Recycle Bin window and pressing delete or you can restore the file to its original location by choosing edit, Undo delete from the Menu bar. Note that files which you delete from a floppy diskette are not placed into the Recycle Bin.

8.9 THE CONTROL PANEL 

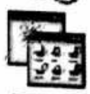
With the Control Panel, you can change the system defaults or properties of your computer. For example, you can customize the way your mouse and keyboard work, Choose **Display** icon and customize the appearance of the desktop using colour schemes, background patterns, settings, screen saver etc.

As the Control Panel window opens, it may look like Figure 8.8. The Control Panel window contains icons depending on the properties you want to view or edit. Once you change the setting with Control Panel, changes are stored. These settings are reloaded each time you run windows and stay in effect until you change them again from the control panel. In your Control Panel the number of items depends on the hardware and the items you opened while installing windows in your system.

Figure 8.8 The **Control Panel** as seen before the **Desktop**

Some of the control panel icons are described in table 8.2.

Table 8.2. Control Panel Icons

NAME	ICON	TASK DESCRIPTION
Accessibility		Allow you to set keyboard, mouse, sound display and other option that make windows easier to use by disabled.
Add New Hardware		Installs or removes new hardware
Add/Remove /Program		Installs or Uninstalls Software programs
Date/Time		Change the current date and time
Display		Sets the colours and fonts of various parts of the windows screens, sets the background pattern or pictures for the Desktop. Also allow you to choose screen saver, display driver, at energy saving mode.
Fonts		Add and delete typefaces for your screen display and view fonts.
Joystick		Configure the joystick
Keyboard		Sets the character repeat rate, cursor blink rate and keyboard language.
Modems		Installs or Uninstalls a modem modifies a modem's properties.

Mouse	Sets the speed of the mouse pointer, double-clicking speed, reverse the functions of the right land left buttons, change the appearance of mouse pointer.
Multimedia	Sets up multimedia hardware components, changes the Audio, MIDI, CD music and other multimedia device drivers, properties.
Network	Add network components, changes how the network identifies your computer, defines access to network resources.
Passwords	Sets up or changes passwords.
Printers	Installs or removes a printer, modify the property settings of the printers.
Regional settings	Configures the number, currency, date and time format for your region of the world.
Sounds	Assign sounds to window events.
System	Displays information about your systems internal devices; RAM size, type of processor etc. Views a list of installed devices, change properties of installed devices, and modify windows performance settings.

8.10 SHUT DOWN PROCESS

It is advisable that you always follow the steps suggest to properly shut down all the running programs before you turn off your computer. This ensures that Windows saves your work on disk and that no data are lost. This is done by closing the Exit command from menu or clicking on the program's **close button** .

In Windows when you click **Start** button on the taskbar, the start menu dialog box appears as shown in Figure 8.9 that contains a pull-up list box. Choose **Shut Down…** option from the commands on

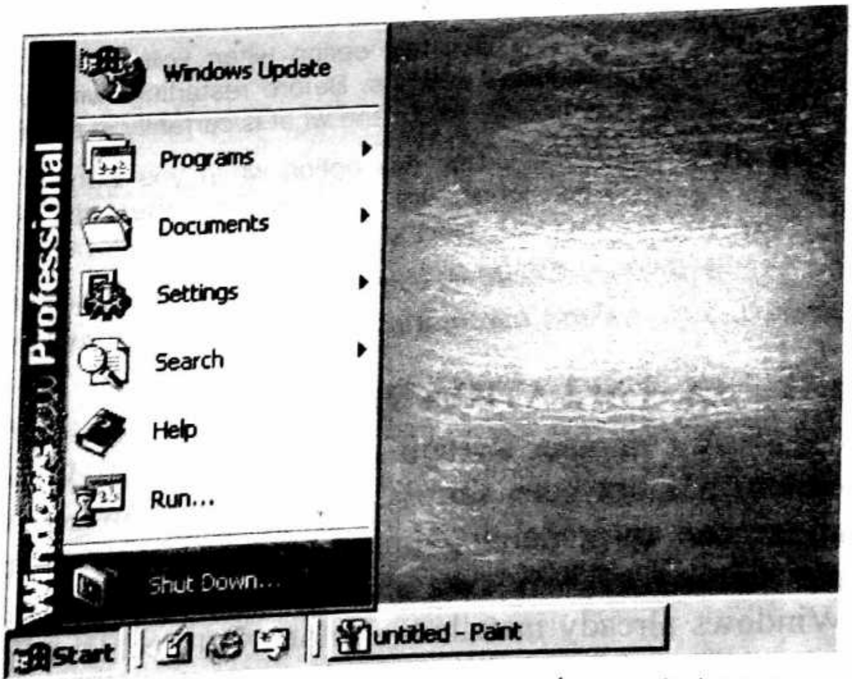


Figure 8.9 Active **Shut Down …** option on start menu

Start menu. Shut down Dialog box as shown in Figure 8.10 appears depending

on the version of Windows installed on your system. You may select an option from the drop-down list by clicking the down arrow [▼] next to the text box as shown in the Figure 8.10. The options contained in the drop-down list are described in the table 8.3.

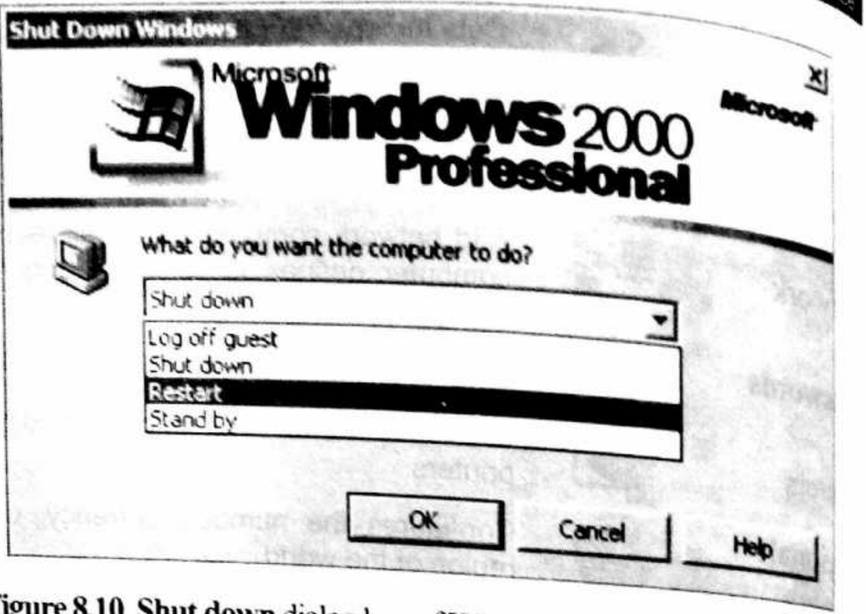


Figure 8.10 Shut down dialog box of **Windows 2000** with **OK/Cancel/Help** buttons. **Restart** option being selected on the pull-down menu list.

Table 8.3 Shut Down Options

COMMAND	WHEN TO USE
Stand by	In stand by mode, computer consumes less electric power but remain available for immediate use.
Shut Down	Use this option when you want to turn off your computer. Windows prepares for your system to be turned off by saving any systems settings that you have made and anything that is currently in RAM to hard disk.
Restart	Use this option when you want to restart your computer with new windows settings. Before restarting, windows save the system settings that you have made and what is currently in the memory to the hard disk.
Log off	Use this option when you want to close all the running programs, disconnect the computer from the network. This option ends your current session, leaving the computer ready for use by another person.

Wait until Windows completely shut down and tells you that Windows is shutting down. Latest systems automatically turn off as soon as Windows shut down completely.

8.11 INSTALLATION OF WINDOWS

A computer system does not respond to its user unless an Operating system has not been installed in it. Windows is an operating system that has undergone up-gradations during the last twenty years. When you are installing a new version of Windows, chances are that your system has some old version of Windows already installed. Before you decide to install **Windows 2000** or any other version of Windows *(Operating system manufacturers are upgrading their Operating Systems on regular basis to meet the user's need)*, you are required to make sure that you computer has a CPU, disk space, and RAM necessary for the

Operating system. Hardware requirement necessary to install **Windows 2000** effectively are:-

- The CPU must be 133-MHz or higher Pentium or Compatible (*Windows 2000 does not run on 486 systems. However you can install older versions of Windows on 486 machines*).
- The minimum amount of RAM is 32 MB, which works but is very slow. Your Windows session gets faster if your system has 64 MB or larger RAM.
- Windows 2000 requires larger space on your hard disk, at least 600 MB free space. (*More space might be needed depending upon the components being installed. The file system used. The method used for installation. And the size of the paging file.*)
- A CD-ROM drive, if you intend to install Windows from CD-ROM. Or a Windows 2000 network adapter.

There are a number of basic steps that must be taken before new installation of Windows. These are :-

- ✓ Prepare back up your current files on a disk or another computer.
- ✓ Uncompress any drive-space or double space volumes before installing Windows unless the drive was compressed with the NTFS file system compression feature.
- ✓ Disabled disk mirroring before running Setup.
- ✓ Disconnect UPS, if you have it with your computer.

Some of the methods for a new installation of Windows are:-

i. From the CD-ROM on a computer running MS-DOS

- Insert the CD-ROM in the drive.
- At the command prompt, type; **D:** **(Where D is the drive letter of CD-ROM drive)**
- Press <**Enter**> ↵ key.
- Type; **CD I386**
- And then press <**Enter**> ↵ key.
- Type; **WINNT**
- And then press <**Enter**> ↵ key.
- Next Follow the Setup instructions.

ii. By starting the computer from floppy disks.

- Locate both the Windows 2000 Setup floppy disks and the Windows 2000 CD-ROM.
- With your computer turned off, insert first Setup disk into its drive A.
- Turn on your computer.
- Next Follow the Setup instructions.

(Setup works in several stages, prompting you to insert disks and the CD-ROM, requesting information, copying files, and restarting. Setup concludes with the Configuration of your system, which you can use to adjust the configuration for your specific needs).

iii. **By starting the computer from the CD-ROM.**

- Locate the Windows 2000 CD-ROM.
- With your computer turned off, insert the CD-ROM into its drive.
- Turn on your computer and wait for Setup to display a dialog box.
- Next Follow the Setup instructions.

8.12 COMPUTER VIRUS

In the past few years, a new processing crime has gained the attention of computer users all over the world. Some people have found a way to create programs that silently replicate themselves on storage media without the computer users realizing it. These programmes are referred to as computer viruses. A computer virus is a program that literally infects other programs and databases upon contact. When the program is loaded into the computer, the virus attaches itself to other programs that are residing in the system. When some one inserts an infected disk into a computer, the files in computer's memory become infected. The reverse is also true, that is, a disk used in an infected computer becomes infected. Computers can be infected when a hacker creates a virus and sends it over the phone lines to a network. Since network is connected to thousands of computers, the infection is carried to all the connected computers. Running a program or accessing the network or infected disk activates the virus without the user's knowledge.

There are many types of viruses. Some act quickly by erasing user programs and files on disk. Others grow like a cancer, destroying small parts of a file each day. Some acts like a time bomb. They lay dormant for days or months and all of a sudden become active, attacking on any software on the system at a particular time. When the virus arrives at a computer, it performs the activities assigned to it for which it was created. Some of the activities that a virus has been programmed to do are:-

- *Copy themselves to other programs.*
- *Display information on the screen.*
- *Destroy data files.*
- *Erase an entire hard disk.*
- *Lie dormant for a specified time or until a given condition is met.*

Unlike the viruses that cause colds and diseases in humans, computer viruses do not occur naturally. A virus program may be harmless, simply producing an obscene or silly message unexpectedly on the computer screen. But

it may also be a very destructive and harmful. It may wipe out huge amount of data or corrupt files on the user's hard disk. Whatever is the motive behind it, anger, revenge, or intellectual challenge, it is clear that people creating virus do not have respect for other people's hard work.

8.13 ANTI VIRUS

Computer users need to protect their computers from viruses until they are eradicated from the source and until they no longer exist. Fortunately there are solutions to safe guard your system against viruses. The first thing that you need to know is the ways in which your system may be infected. The most common ways that may infect your system are:-

i. *A floppy or a removable hard disk from another user may be a carrier of virus.*

ii. *Virus may be transmitted to other computers via Network. That is your system may be infected when you link your system with another user, an on line service, or the Internet.*

iii. *Programme purchased from unauthorized dealers may be the potential carrier of viruses.*

A number of programs have been designed to protect software from viruses. These are called anti-virus program. However no anti-virus program guarantee protection against all viruses. If you suspect that a program is a carrier of virus, you can analyse the program using anti-virus program to see if it contains any know viruses. Most anti-virus programs can be used to eliminate a virus found in a disk. Many computer users have installed these anti virus programs on their computers. A good anti-virus program checks for infected files automatically every time you insert any kind of disk or use your modem to retrieve a file. Several anti-virus programs are available and some are even free. Some common anti-virus programs are:-

- McAfee Virus scan
- Norton Antivirus
- Symantec Antivirus
- Virex
- Disinfectant

SUMMARY

Windows is a **graphical user interface** (GUI) and is easy to learn. Once **Windows** is loaded, it takes **charge** until you shut down the computer.

THE DESKTOP

The Desktop appears when you load Windows. This contains small pictures or graphic objects called **icons** and a bar at the bottom.

ICONS

The small graphic objects such as **My Computer**, **Recycle Bin** etc. seen on the Desktop are called Icons. Icons carry out specific task on the desktop.

BUTTONS

There is another class of symbols that is used in a GUI called buttons. Buttons are the areas of screen where clicking causes something to happen.

KEYBOARD

Keyboard is an input device for entering text. It also offers shortcut methods for performing commands and procedures.

MOUSE

An arrow seen on the screen is a mouse pointer. Some mouse actions in Windows are Pointing Clicking, Double-clicking, Right-clicking and Dragging.

LOADING WINDOWS

When you turn on your computer, Windows is loaded automatically into the computer's memory (RAM) from the hard disk. This is called booting.

DESKTOP

Windows desktop includes number of icons on the left and a horizontal bar normally positioned at the bottom of the screen. This bar is called the taskbar.

THE START BUTTON

Start button is located on the left end of the task bar. It activates the Start **Menu** through which Windows offer everything installed on your computer.

TITLE BAR

The title bar at the top of a window contains name of the application and a set of control **icon** having **Minimize, Maximize/Restore and Close** Buttons.

STATUS BAR

The **status** bar provides useful information about the window

SCROLL BARS

When there is more information then **scroll** bars are displayed at the right or bottom borders. **scroll** bars move a window by clicking the **arrow-heads**.

MENU BAR

The **Menu** bar appears immediately below the **title** When **Menu** bar is activated, a **drop-down menu** appears below it that lists all its commands.

MY COMPUTER

My computer allows access to computer resources from hardware devices to file folders such as Floppy drive, Disk Drive, Control Panel, etc.

THE RECYCLE BIN

Any item deleted from the hard disk goes to **Recycle bin**. The Recycle bin temporarily holds the deleted item until it is removed permanently or restored.

THE CONTROL PANEL

With the Control Panel, you can change settings of your computer such as Date/Time, Display, Fonts and Mouse etc. These changes are stored and are reloaded each time you run Windows and stay until you change them.

SHUT DOWN PROCESS

Close all the running programs using Exit command from the menu or clicking on the program's **close button** before you turn off your computer. This ensures that Windows saves your work on disk. Choose **Shut Down...** option from the **Start menu**. Select **Shut down** option and click OK in the dialog box.

INSTALLATION OF WINDOWS

Before you decide to install **Windows 2000** or any other version of Windows, make sure that you computer has the CPU, disk space, and RAM necessary for its installation.

COMPUTER VIRUS

A virus is a parasitic program that is attached to a computer program or file through an infected disk or phone lines to a network. Through the network, the infection is carried to all the connected computers. Running an infected program activates the virus without the user's knowledge. When a virus arrives at a computer, it performs the activities assigned to it. It may be very destructive and may wipe out huge amount of data or corrupt files on the user's hard disk.

ANTI-VIRUS

Computer users need to protect their computers for viruses. The first thing that you need to know is the ways in which your system may be infected.

A number of programs have been designed to protect software from viruses. These are called anti-virus program. Anti-virus programs are used to eliminate a virus found in a disk. However no anti-virus program that are available guarantee protection against all viruses.

EXERCISE

8.01 Complete the following statements

i) DOS is a ________________________ interface.

ii) Windows is a ____________________ interface.

iii) The icons on the desktop allow you to view and ________ with Windows.

iv) The ______________ button has both name and icon.

v) To close DOS program type ____________ at the DOS prompt.

8.02 Tick (✓) the following statements either True or False.

i) Menu bae appears just below the status bar. True/False

ii) Status bar shows more than six commands on it. True/False

iii) Screen saver can blank your screen when you are not using mouse or keyboard. True/False

iv) Any file you delete from the hard disk, goes straight to Recycle bin True/False

v) The icon  on the Control Panel is used to install uninstall software program. True/False

8.03 Encircle one choice A, B, C or D in each of case.

i) Any command that is not available for selection in the drop-down menu list appears:-
(A) Dark (B) Invisible (C) Dim (D) Underlined

ii) The number of mouse actions with two button mouse are:-
(A) 3 (B) 4 (C) 5 (D) 6

iii) A textbox is a type of :-
(A) Command box (B) Menu box(C) List box (D) Dialog box

iv) Number of Shut down options on Windows 2000 professional are :-
(A) 2 (B) 3 (C) 4 (D) 5

v) The commands that is not present on edit menu?
(A) Font (B) Go To (C) Find (D) Ctrl + Z

vi) The number of items present on Control Panel depends upon

(A) Hardware items attached with the system
(B) Items contained on Windows CD
(C) Item installed during Set up
(D) Item required by your system.

8.04 Match the items given in Column I with those in Column II

	Column I		Column II
i)	Toolbars	a)	Replication
ii)	Ctrl + Z	b)	Table menu
iii)	Ctrl + V	c)	Tools Menu
iv)	Disinfectant	d)	View Menu
v)	Computer Virus	e)	Undo action
vi)	Office clipboard	f)	Paste Command
vii)	Page Setup	g)	File Menu
viii)	Ruler	h)	Copy Command
ix)	Ctrl +C	i)	Edit menu
x)	Sort	j)	Anti Virus Program

8.05 What is an operating system?

8.06 Describe three primary mouse movements in Windows?

8.07 Describe four common mouse pointer shapes?

8.08 What does a window represents in a GUI?

8.09 What happens when you maximize a window?

8.10 What are two important characteristics of a well designed GUI?

8.11 What is meant by plug and play?

8.12 Describe the purpose of the task bar?

8.13 What is meant by the phrase **wallpaper your desktop**?

8.14 What is the benefit of using a screen saver?

8.15 Compare MY COMPUTER and WINDOWS EXPLORER?

8.16 What is a Computer Virus? Describe some of their activities. How computer virus can be checked in your disk?

9

PROBLEM SOLVING

Every day you have to solve number of problems of different nature. Many problems have more than one solution. Often you solve them without considering all the steps involved. For example to see a friend at his house involves decisions such as when to leave; the transport you need; and the route to follow. **To reach at a solution, any number of different decisions could have helped you.**

Computer problem solving is similar. Many solutions may exist for a single problem. Some may be more efficient than the others. Only a **well-planned program** can properly direct a computer to solve a problem correctly.

9.1 PROBLEM SOLVING ON COMPUTERS

Problems can be solved on computers. It requires knowledge to develop suitable software necessary for their solutions. To solve a problem correctly and efficiently require a computer program that has been developed after detailed planning. **Computer Programming** can be best described as **problem solving**. There are **two crucial steps** in the problem-solving process:-

- *To ensure that the correct problem is being solved and*
- *To develop a correct logic to solve the problem.*

The object is to produce a **Complete, Correct and Understandable** solution of a problem. Example 9.1 illustrates various stages of problem solving.

EXAMPLE: 9.1

Suppose you **need** a **calculator** and you **want** to **buy** it.

Here two important points are already clear to you, that is, you **need a calculator** and you **want it to buy**. You are further require to know whether you need a simple calculator or a scientific calculator, and should it be a table type or a pocket calculator? As you need it to carry with you so your choice is a pocket size scientific calculator. This step **defines the problem**.

Next requirement is to know the performance of various makes/models and the prices of those calculators. You are also required to know when and from where you have to buy it. This is how the **problem is analyzed**.

Now you prepare a plan to buy a calculator. If someone else is buying it for you, then you need to **write** each step of your plan clearly so that it could convey him the necessary detail. You must also **test** how correctly your plan (**program**) be interpreted. And **correct** it, if you find any error.

Finally is the **implementation** stage. For this you select a suitable time and take reasonable money to buy the calculator. Visit the shops where such calculators are available. Get the one you require, check its working and pay its price.

It is clear that you reach at a proper solution when all the steps are taken correctly. A programmer follows similar steps when solving problems on computers.

9.2 PROGRAM PREPARATION

A computer cannot make unplanned decisions. Good programming means that various steps leading to proper solution of a problem must be understood clearly. Generally these include the following stages:-

1. **Problem Identification stage.**
   a) Defining the Problem. (Knowing the objective of the problem)
   b) Analyzing the problem. (Understanding the problem).
2. **Planning Stage.**
   a) Developing the algorithm. (Step by step problem solving procedure).
   b) Flow charting. (Pictorial view of the algorithm)
   c) Writing Pseudocode. (Writing the algorithm in a simple language)
3. **Coding and Testing stage.**
   a) Writing the program. (Converting the algorithm into a Computer program using a programming language)
   b) Testing and debugging the program. (Testing and removing errors in the program)
   c) Running the program. (Using the program)
4. **Implementing and Documenting stage.**
   a) Implementing the program. (Taking the program and placing it into operation)
   b) Documenting the program (Describing the input, the output it produces and the way the computer must be operated)

The key to good programming is planning. By spending more time in the planning phase, it is normally possible to save time in **writing, testing** and

implementing a program. In many cases better planning can reduce total programming time and cost. This concept is shown in Figure 9.1.

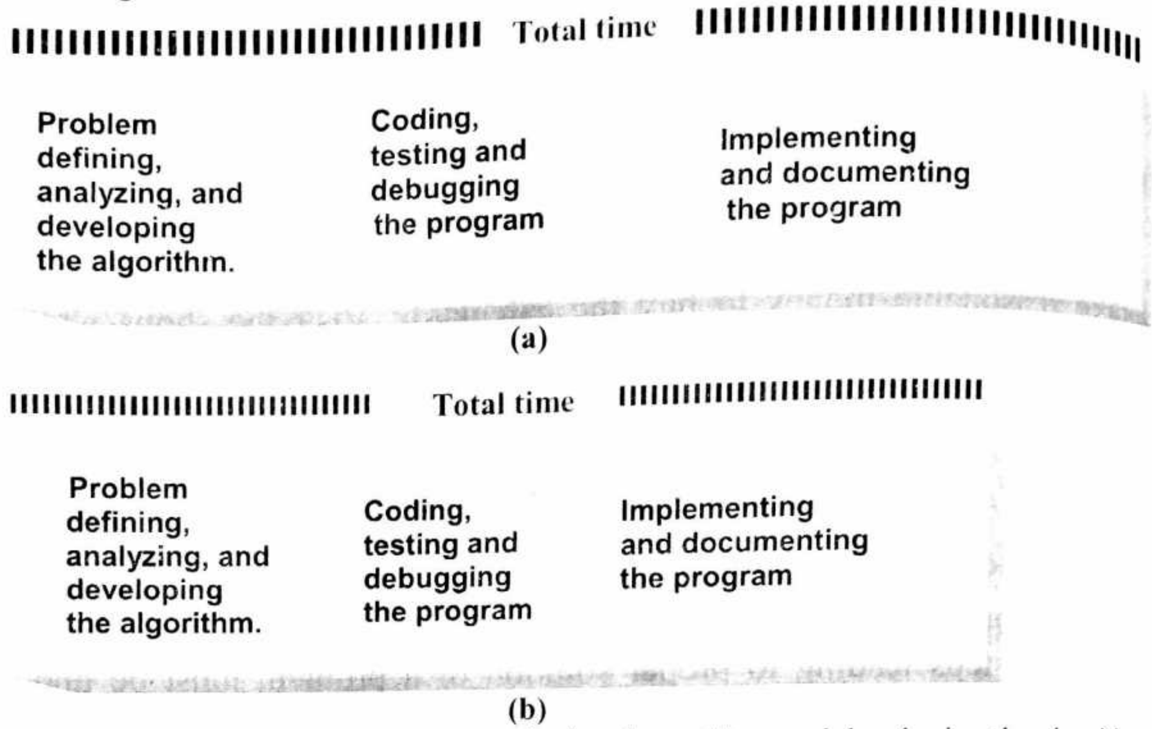


Figure 9.1 The importance of defining, analyzing the problem and developing the algorithm.

9.3 PROBLEM IDENTIFICATION STAGE

The first stage of program development is to identify the problem. This is an important stage in which we **define** and **analyze** the problem to understand it properly because only then we can write a proper program.

DEFINING THE PROBLEM

Often only a verbal statement of the objective is enough. But sometimes, it may be useful to write down the objectives and other important factors of the problem. Problem defining step should include identification of the input and the desired output of the program. To define a problem we proceed as follows:-

- Study the problem carefully.
- Is the solution possible?
- Identify inputs and outputs of the problem.
- Note the important conditions necessary to solve the problem.

ANALYZING THE PROBLEM

An important step in developing a solution to any problem is the process of understanding the problem called the analysis of the problem. A series of questions can be asked to analyze a problem such as:-

- *Do we know how to solve the problem?*
- *Can the problem be solved on computer?*
- *What are the inputs and outputs?*
- *How many solutions are possible?*
- *Which one is the simplest solution of the problem?*

Let us study the following examples.

EXAMPLE 9.2

Suppose **you** want to consult a doctor for treatment.

SOLUTION:

Defining the Problem

To **visit** the clinic of (*certain*) doctor on (*date*) at (*time*) **for consultation**.

(This defines the Problem. That is, what to do? When to do? And what are the objectives?)

Analyzing the Problem

Now let us analyze the problem by asking the following questions:-

- Who will accompany you, when you visit the clinic?
- What are the clinic timings?
- How far is the clinic from your home?
- Do you need any transport?

(The problem may be **analyzed** after getting answers of these questions).

EXAMPLE 9.3

How will you find the average of three numbers?

SOLUTION:

Defining the Problem

The statement "to find the average of three numbers" **defines** the problem.

Analyzing the Problem

To analyze the problem you may require answering the following questions.

- Do you know how to find the average of numbers?
- Is the solution possible on computer?
- Can you plan to solve the problem on computer?

9.4 DEVELOPING THE ALGORITHM

Once a programmer understands the problem, next stage is the planning stage. The planning stage consists of developing an **algorithm** and converting it into a **flow chart** or **pseudocode** or both. **Algorithm** is a **step-by-step procedure**

developed to solve a problem before writing an actual program. More precisely an **algorithm** is a complete procedure or plan that describes the logic of a program.

In any situation programmer would usually have a choice of algorithm. Each method will have its own merits and demerits. One of the important job of a good programmer is to select the best one with reasons of selecting that. In order to qualify as an algorithm, it should possess the following characteristics:-

- Each and every step should be precise and clear.
- Each step should be performed in a finite time.
- Steps should not be repeated infinitely.
- The desired result should be obtained after the algorithm terminates.

For computers, algorithms must be very precise. Strange things may happen when ***a computer does what the programmer tells it to do rather than what the programmer wants to do****. There are lots of jokes about recipes which illustrate* the **Cook with blue face and teeth chattering** *who followed the instructions* "**stir the mixture well and stay in the fridge for 5 minutes**". *Unfortunately, there is no quick and easy way to learn this skill. The best way, as usual is to start with the simple example; further, you must* ***learn by doing*** *rather by watching to develop algorithm yourself.*

EXAMPLE 9.4

Develop an algorithm to prepare tea.

SOLUTION:

Defining the Problem

The statement "to prepare tea" **defines** the problem.

Analyzing the Problem

To analyze the problem you may require to answer the following questions:-

i) Do you know how to prepare tea?
ii) Do you have the necessary ingredients?
iii) Do you have the necessary apparatus for its preparation?

After defining and analyzing the problem, we develop an algorithm as under:-

Algorithm

Step 1: Take water in the tea-pot.
Step 2: Heat it until the water begins to boil.
Step 3: Put tea leaves in the boiling water and stop heating.
Step 4: Cover the tea-pot and wait for 10 minutes.
Step 5: Take some mixture from tea pot into a cup.
Step 6: Add some sugar and milk into the cup and stir it with a spoon.
Step 7: Tea is ready. Stop.

EXAMPLE 9.5

Develop an algorithm for the problem described in example 9.3 in which you are required to find the average of three numbers.

SOLUTION:

In the **Planning stage,** we design a step-by-step that may be as follows:-

Algorithm

Step 1: Represent the three numbers by A, B and C.
Step 2: Add the three numbers A, B and C to find their Sum.
Step 3: Divide the Sum by 3 to get the average of A, B and C.
Step 4: Stop.

EXAMPLE 9.6

You are given result cards of 30 students of class X . Write an algorithm to find out numbers of students placed in grade **A1**, **A** and **B** respectively.

SOLUTION:

Defining the Problem

Finding out the number of students of class X placed in grades A1, A and B.

Analyzing the Problem

- Given, result cards of 30 students placed in various grades.
- To find how many of them are placed in grade A1, A and B

After defining and analyzing the problem, we develop the algorithm as under:-

Algorithm

Step 1: Initialize TOTAL CARDS CHECKED, TOTAL AI, TOTAL A, and TOTAL B grades equal to zero.
Step 2: Take result card of a student.
Step 3: Check the grade on the result card. If it is A1, then add 1 in the number of TOTAL A1 and go to step 6, if not then go to step 4.
Step 4: If the grade on the result card is A, then add 1 in the number of TOTAL A and go to step 6, if not then go to step 5.
Step 5: If grade is B, then add 1 in the number of TOTAL B.
Step 6: Add 1 in TOTAL CARDS CHECKED.
Step 7: Is TOTAL CARDS CHECKED = 30, if not then take the result card of next student and go to step 3.
Step 8: Print TOTAL A1 grades.
Step 9: Print TOTAL A grades.
Step 10: Print TOTAL B grades.
Step 11: Stop.

EXAMPLE 9.7

Find the area of a triangle whose sides are a, b and c.

SOLUTION:

Defining the problem

The inputs are the sides a, b and c of the triangle and area as its output.
Is it possible to determine area of the triangle? *Yes, it is possible.*

Analyzing of the problem

We know that the area (say A) of a triangle, when all its sides a, b, and c are given, can be found using Hero's formula.

Developing the Algorithm

Step 1: Sides of the triangle are a, b and c

Step 2: Sum of the sides = $a + b + c$

Step 3: Then $s = \dfrac{a+b+c}{2}$

Step 4: $\therefore$ Area of the triangle $A = \sqrt{s\,(s-a)\,(s-b)\,(s-a)}$

9.5 FLOW CHARTING

Once we have an algorithm, we are well on the way to write a program. However we need to go one step further to ensure that the program we write will actually represent the algorithm stated previously. For this the algorithm is expressed in one or both the following forms:-

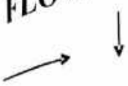
- The **pictorial form** is referred as a **flowchart**, *which is a pictorial representation of an algorithm.*
- The **written form** of the algorithm (not in computer language) is called **pseudocode**. *Pseudocode expresses the basic logic structure of an algorithm in a systematic and clear way.*

A **flowchart** is a pictorial view that illustrates the algorithm. Flowcharts are perhaps the best method for expressing what the computer is going to do. The flow chart helps us in planning work in an organized manner. It demonstrates clearly the logical flow of the computer program. A flowchart can help the persons, the one who is running the program and someone else who is reviewing the program.

FLOWCHART SYMBOLS

A standard set of flowchart symbols is used to illustrate the algorithm. The actual flow of control is represented by arrows which connects the symbols. Some of the symbols used to draw a flow chart are described here:-

FLOW LINE

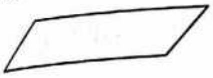


A line with an arrow head represents the flow of control between various symbols in a flowchart.

TERMINAL SYMBOL

An oval shape symbol is used to represent the starting and the stopping point of a flow chart. The following terminal symbols indicate the beginning and ending of the flow chart.

INPUT/OUTPUT SYMBOL

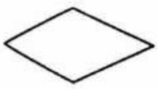


A parallelogram represents either input or output operation regardless of the input or output method. The following two symbols shows that the values X and Y are input. The third symbol shows the value X as output.

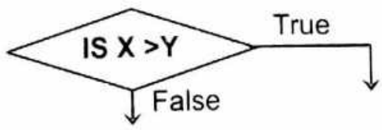
PROCESSING SYMBOL

A rectangular block is used to represent processing symbol. It shows all the calculations such as given below.

DECISION SYMBOL



A diamond represents a decision symbol used for comparison or a decision. It changes the flow of control and computer decides a particular path to be followed. The decision box has at least two exits. Given below a logical decision, whether **X** is greater than **Y** or not.



CONNECTOR SYMBOL

A small circle represents a connector symbol and is used to join various parts of a flow chart. Connectors are used when a flow chart is very large and the numbers inside them identify their links. Control is transferred from one connector to another with the same number in a program.

Let us now apply the concept of flow chart in examples 9.2, 9.4 and 9.5

DEVELOPING FLOW CHARTS

EXAMPLE 9.8

Develop a flow chart expressing the algorithm described in example 9.2

Figure 9.2 shows a flow chart that describes the procedure to visit the clinic to consult the doctor according to the algorithm developed for example 9.2

Figure 9.2 A Flowchart showing the procedure for consulting the doctor

EXAMPLE 9.9

Develop a flow chart for example 9.4 that describes the preparation of tea.

Figure 9.3 shows the flowchart for preparing tea according to the algorithm developed for example 9.4

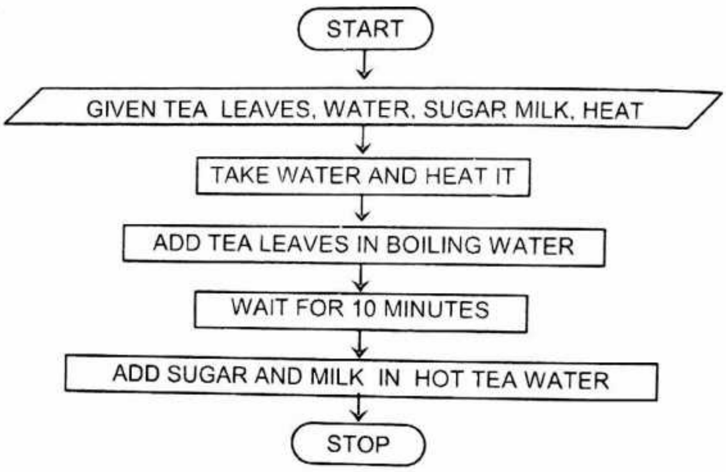


Figure 9.3 Flowchart for preparing Tea

EXAMPLE 9.10

Develop a flowchart expressing the algorithm described in example 9.5.

Figure 9.4 shows a flowchart that describes the procedure to find average of three given numbers according to the algorithm developed for example 9.5

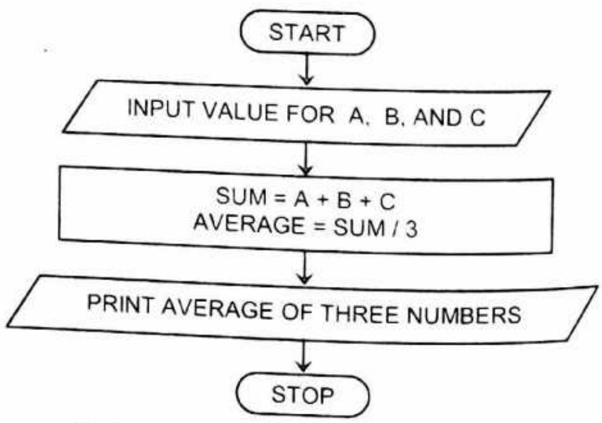


Figure 9.4 A Flowchart showing the procedure to find the average of three numbers

Let us consider some more examples from daily life and develop flowcharts that illustrate the procedure to achieve the objectives.

EXAMPLE 9.11

Develop a flow chart to illustrate how to contact to your friend on telephone.

SOLUTION

1. *The first step is to check whether the telephone set is in* ***working order****.*
2. *Then the next step is to* ***dial the number*** *of your friend.*
3. *After the number being dialed,* ***note the response*** *from the other end.*
4. *If the number is* ***busy*** *then return to the* ***"wait"*** *process.*
5. *Dial the number again after some time and* ***wait for reply****. If no reply, then there may not be anybody to attend the call.*
6. ***Put down the telephone set.***
7. *If somebody replied then* ***inquire for your friend****.*
8. *If your friend is* ***not available*** *then* ***leave the message*** *for him.*
9. *If* ***available*** *then* ***talk*** *to him. The complete flow chart is illustrated in Figure 9.5*

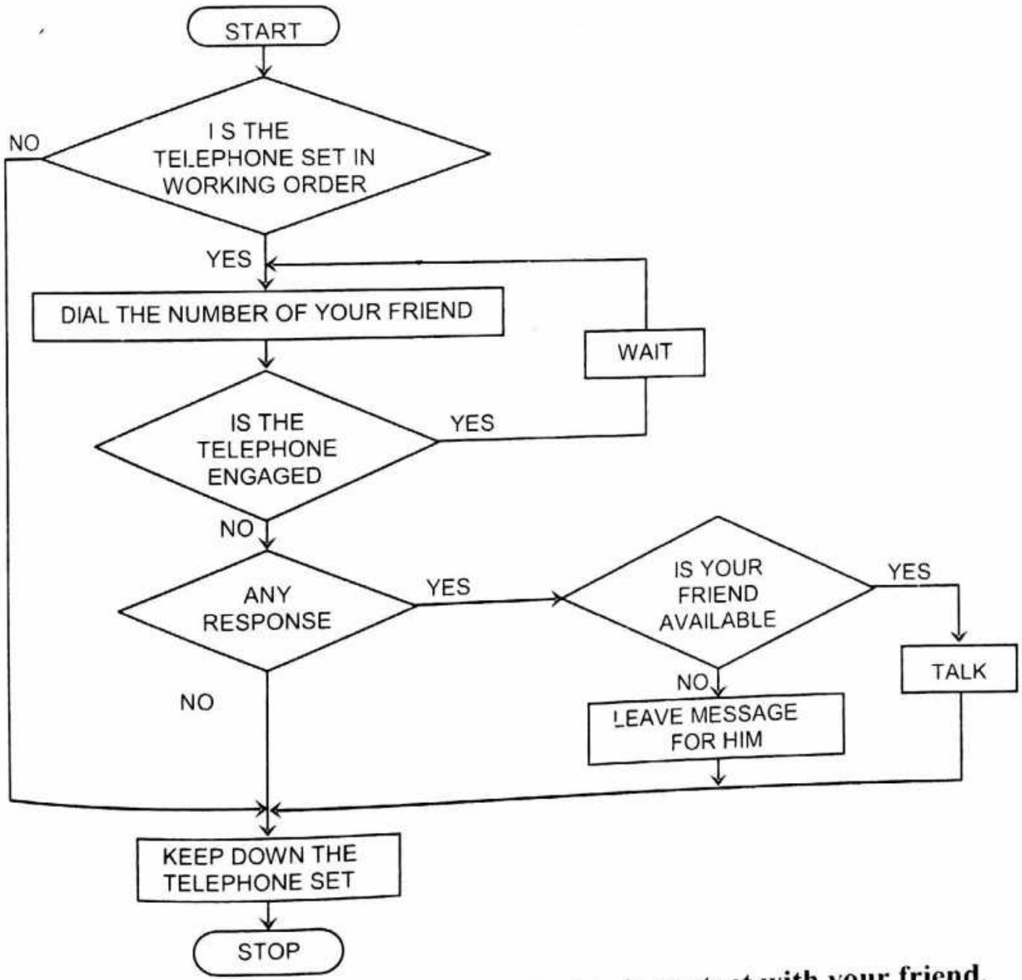


Figure 9.5 Flow chart for a telephonic contact with your friend.

9.6 CODING THE PROGRAM

The process of converting the algorithm into a **computer program** using a language that a computer can understand is called **coding the program** or **writing the program**. The success of this step greatly depends upon the quality of previous steps.

A computer program consists of series of instructions. All these instructions may be numbered and the computer executes these instructions in ascending order. A computer follows these instructions to solve the problem and produces the desired result, when executed successfully. Computer instructions are of the following five types:-

Input/Output: These instructions direct the computer to move information to and from the computer's memory and an input or output unit.

Control: These instructions control the order in which other instructions are executed by transferring the control to the instruction executed next.

Arithmetic: These instructions direct a computer for arithmetical computations and moving data from one place to another in memory.

Logical: These instructions enable the computer to compare items of data and proceed according to the result of the comparison whether certain condition is **true** or **false**.

Specification: These instructions are descriptive. A programmer can inform a computer about things such as the types of data items used in a program, the allocation of storage so on and so forth.

9.7 BASIC, THE PROGRAMMING LANGUAGE

Every language has its own syntax (grammar) and vocabulary. The language, which is the most commonly used by the students or by the owners of simple personnel computers as programming language, is BASIC (**B**eginner's **A**ll purpose **S**ymbolic **I**nstruction **C**ode).

BASIC is easy to learn, easy to use language. Many of the terms in BASIC are based on English words. The language uses simple English Key words like INPUT, READ, PRINT, GO TO, IF, THEN, ELSE etc.

This language is used for solving large variety of problems relating to business, medicine, science and engineering. The simplicity and versatility of BASIC has made it very popular language.

EXAMPLE 9.12

Consider a simple problem to add and print the sum of three numbers A, B and C. Figure 9.7 shows the flow chart for their sum and to print their sum.

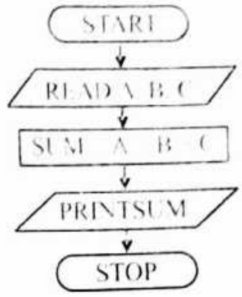


Figure 9.7 Flow chart for the sum of three numbers.

BASIC program related to the flowchart Figure 9.7 is as under:-

```
10   DATA A, B, C
20   READ A, B, C
30   SUM = A + B + C
40   PRINT SUM
50   STOP
```

9.8 TESTING AND DEBUGGING THE PROGRAM

Testing and debugging are very important steps in developing computer programs. Errors or bugs in computer programs are common. The process of finding and removing errors from a computer program is called **debugging**.

Once the program has been written, it must be tested to ensure that it works correctly and will perform as intended. If a program or module does not produce correct result, it must be examined for errors, correct the errors, and test the program again. The errors present in a program are generally of the three types; Logical Errors, Syntax Errors and Execution Errors.

LOGICAL ERRORS

Logical errors may be due to improper use of the formula or incorrect formula or wrong use of the symbols or data for example, if we write the formula for speed as:

Speed = Distance * Time

Instead of Speed = Distance / Time

Similarly incorrect formula for average of three numbers A, B and C as:

Average = A + B * C / 3

Instead of correct formula as:

Average = (A + B + C) / 3

A computer cannot detect such errors and thus give out wrong results. Such types of errors are called logical errors. To detect logical errors, we run the

program with test data. When a logical error is found, it is removed by making necessary changes in the program.

SYNTAX ERRORS

Syntax errors caused due to the wrong use of a programming language. This involves incorrect punctuation, incorrect word sequence, undefined terms or misuse of the terms. These errors are automatically pointed out by the language processors. For example if a statement such as

$$C = A + /B$$

appears in the source program, then the computer will not process the data. The computer will indicate the syntax error on execution.

EXECUTION ERRORS

Execution errors arise due to the limitation of the computer. For example if a program contains a division of a number by zero. In such cases either the execution of the program is held up or sometimes it gives absurd results. The execution errors like logical errors are also difficult to locate.

9.9 RUNNING THE PROGRAM

After the program has been executed correctly with test data, the program is ready to run and can process the data. At this stage, it is necessary to know whether the results obtained meet the objectives. If a program does not meet the user's needs, then the programmer should trace through the program development procedure and change the algorithm, flowchart, pseudocode and the program coding.

EXAMPLE 9.13

Let us write a program for the addition of three numbers as shown below:-

```
10 DATA 7, 8, 6
20 READ A, B, C
30 SUM = A + B + C
40 PRINT SUM
50 STOP
```

Program for the sum of three numbers.

Type RUN and press <ENTER> key. The final output so obtained is shown below:-

```
21
```

Output of the sum of three numbers (7, 8 and 6).

9.10 IMPLEMENTATION

Implementation is a process of taking the program and placing it into operation. One of the best ways to implement a new program is called running in parallel. Under this condition, the new programs run with the existing system. If there are problems with the new programs, they can be corrected while the existing system is still being used. After the bugs are out, the new programs are slowly **phased in**, while the old system is slowly **phased out**.

9.11 DOCUMENTATION

A program may be capable of fulfilling the task successfully for which it has been designed. But it may be useless if it is not well documented that tells the user about the input necessary for the program, the output it produces and the way the computer must be operated.

Well-documented programs are extremely valuable whenever the program is to be rewritten for another computer or whenever someone else wants to modify the program. For most applications, two different types of documentation are required. These are **Users documentation** and **Technical documentation**.

USER DOCUMENTATION

User documentation is developed in easy-to-understand terms for the individuals who use the program. Although there is no standard way to develop user documentation, the following material may be included:-

- A discussion of the problem that will be solved by the program.
- A non-technical discussion of how the problem is solved by the computer.
- A description of the output from the program.
- A discussion of the data that is required to run a program.
- A list of potential problems that may be encountered by the user.

TECHNICAL DOCUMENTATION

Technical documentation helps the computer operators to execute the program. It is also used by the analysts and programmers in case there are problems with the program or if the program needs modification. This type of documentation should include the statement of the problem and all structure charts, flow charts coding forms, and other material developed during the problem definition, analysis and design. In addition technical documentation can include a list of potential problems and a description of how to solve such problems.

SUMMARY

In computer programming only a **well-planned program** can properly direct a computer to solve a problem correctly.

PROBLEM SOLVING ON COMPUTERS

Problems can be solved on computers. It requires knowledge to develop suitable software necessary for their solutions. To solve a problem correctly and efficiently require a computer program that has been developed after detailed planning. **Computer Programming** can be best described as **problem solving**.

PROGRAM PREPARATION

A computer cannot make unplanned decisions. Good programming means that various steps leading to proper solution of a problem must be understood clearly. Generally these include the following stages:-

Problem Identification stage, Planning Stage, Coding and Testing stage, Implementing and Documenting stage.

The key to good programming is planning. By spending more time in the planning phase, it is normally possible to save time in **writing, testing** and **implementing** a program.

Problem Identification Stage

The first stage of program development is to identify the problem. This is an important stage in which we **define** and **analyze** the problem to understand it properly.

Developing an Algorithm

Once the programmer clearly understands the problem, the next stage is the planning stage. The planning stage consists of developing algorithm and the conversion of algorithm into a flow chart or pseudocode or both.

Flow Charting

A **flowchart** is a pictorial view that illustrates the algorithm. The flow chart helps us in planning work in an organized manner. It demonstrates clearly the logical flow of the computer program. A flowchart can help the persons, the one who is running the program and someone else who is reviewing the program.

Flowchart Symbols

A standard set of flowchart symbols is used to illustrate the algorithm. Some of the symbols used to draw a flow chart are terminal symbol, input/output symbol, processing symbol, decision symbol and connector symbol.

CODING THE PROGRAM

The process of converting the algorithm into a computer program using a language that a computer can understand is called coding the program or writing the program.

A computer program consists of series of instructions. A computer executes these instructions in a sequence. Computer instructions are of the following five types:- Input/Output, Control, Arithmetic, Logical and Specification instructions.

BASIC PROGRAMMING LANGUAGE

The programming language, which is the most commonly used by the students or by the owners of simple personnel computers is BASIC (Beginner's All purpose Symbolic Instruction Code). BASIC is easy to learn, easy to use language. Many of the terms in BASIC are based on English words.

TESTING AND DEBUGGING A PROGRAM

The process of finding and removing errors from a computer program is called **debugging**. Once the program has been written, it must be tested to ensure that it works correctly and will perform as intended. The errors present in a program are logical errors, syntax errors and execution errors.

RUNNING THE PROGRAM

After the program has been executed correctly with test data, the program is ready to run and can process the data. If a program does not meet the user's needs, then the programmer should trace through the program development procedure and change the algorithm, flowchart, and the program coding.

IMPLEMENTATION

Implementation is a process of taking the program and placing it into operation. One of the best ways to implement a new program is called running in parallel. Under this condition, the new programs run with the existing system. Problems in new programs can be corrected while the existing system is still being used.

DOCUMENTATION

Documentation describes what the program should do. What data is needed, how data is identified and how the output is formatted. Well-documented programs are extremely valuable whenever the program is to be rewritten for another computer or whenever someone else wants to modify the program.

EXERCISES

9.01 Complete the following statements

i) BASIC executes in both the____________ and ____________ mode.

ii) Commands are executable in________________ mode.

iii) A __________ Symbol is used for the start of the flowchart.

iv) _________ shows the direction of flow of control in the flowchart.

v) A flowchart is a pictorial representation of an ______________.

vi) The process of finding and removing errors from a computer program is called _______________.

9.02 Tick (✓) the following statements either True or False.

i) Computer programming can be best described as problem solving. True/False

ii) A parallelogram represents a processing symbol in a flow chart. True/False

iii) The statement "IS A=A+1" is placed in processing symbol. True/False

iv) Debugging is an important part of Planning stage True/False

v) BASIC operates both in interpreter and compiler mode. True/False

vi) Key to good programming is testing/debugging True/False

vii) A computer cannot detect logical errors. True/False

9.03 Encircle one choice A, B, C or D in each of the Multiple choice questions.

i) First generation languages use:-
(A) Pseudo code (B) Binary code
(C) Mnemonic code (D) Decimal code

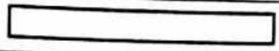
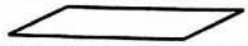
ii) In which of the flowchart symbols, the statement "IS A > B" is placed?
(A) Connector (B) Rectangular (C) Parallelogram (D) Diamond

iii) Which of the following is not a part of planning stage?
(A) Developing algorithm (B) Flowcharting
(C) Coding the program (D) Writing pseudocode

iv) Which types of instructions direct the computer to move information between computer's memory and I/O unit?
(A) Control (B) Input/Output (C) Logical (D) Specification

v) Name the error which you think to exist in the formula C = C*D/C
(A) Logical (B) Syntax (C) Execution (D) None

9.04 **Match the items given in Column I with those given in Column II**

	Column I		Column II
i)	PRINT NUM	a)	Debugging
ii)	Average = A+B+C/3	b)	
iii)	Removing errors of a program	c)	Logical error
iv)	A + ^2	d)	Compilation
v)	Writing Program in HLL	e)	Syntax error
vi)	K = K + 2	f)	
vii)	Translation of High level language into Machine code	g)	
viii)	IS A > B	h)	Coding

9.05. Explain the importance of problem definition in computer with some suitable example.

9.06 Describe the various steps necessary to solve a problem.

9.07 What is an algorithm? What are the characteristics necessary for a set of instructions to qualify as an algorithm?

9.08 Write the algorithm for the conversion of temperature from Celsius scale to Fahrenheit scale.

9.09 What is a flow chart? What are its advantages?

9.10 Describe four symbols used in flow chart.

9.11 Convert the algorithm of question 9.08 into a flowchart.

9.12 Your table lamp is not in working order. Develop a flow chart for locating the trouble in it. The trouble may be either in the plug or in the connecting wire or the bulb itself may be fused.

9.13 Develop a flowchart for identifying and printing an odd number.

9.14 Develop a flowchart for finding printing the sum of squares of all the natural numbers between 2 and 10.

9.15 What is meant by a computer program?

9.17 Mention the advantages of programming in BASIC.

9.18 What is debugging? Why it is necessary to test or debug a program?

9.19 Mention some of the errors that occur in computer programming.

9.20 Why it is necessary to document a program?

10

DATA TYPES, ASSIGNMENT AND INPUT/OUTPUT STATEMENTS

BASIC is an easy high-level computer programming language. It was developed in 1964 at Dartmouth College by Professors John G. Kemeny and Thomas E. Kurtz. Because of its simplicity and conversational nature, BASIC quickly became one of the most popular programming languages.

10.1 BASIC PROGRAMMING

BASIC is the first high level languages to be used on micro-computers. First interpreter of BASIC language was written in 1975 by Microsoft. At one time, there were many versions of BASIC. Most popular versions are GW-BASIC and QBASIC which are included with the DOS operating system.

BASIC has the following features common in all versions:-

- It is an easy, student's friendly language.
- It is a general purpose language, suitable for scientific and commercial programming.
- It has a simple syntax rules and is easy to test and debug.
- Communication with computer is simple and straightforward with it.
- It is well suited for computers having limited memory.

10.2 CHARACTER SET IN BASIC LANGUAGE

Character set in BASIC language consists of alphabetic characters of English language, numeral characters, special characters, arithmetic, relational, logical and functional operators. These are:-

- The **alphabetic characters** in BASIC are capital letters A,B,C, ..., Z as well as small letters a, b, c, ... , z.
- The **numeric characters** are the digits 0, 1, 2, ... , 9.

- **Special characters** such as $, #, %, @, \, blank spaces and punctuation marks ! , : ; ? etc. Some of them have specific meaning in BASIC.
- **Arithmetical, relational operators** such as + - * / ^) (= > < etc.

10.3 RESERVED WORDS

Certain words have special meaning in BASIC. These words are called reserved words. Reserved words cannot be used as variables. There are nearly 160 reserved words in GW BASIC including those given in table 10.1

TABLE 10.1 BASIC Reserved Words

ABS	DEF	GOSUB	MERGE	PRESET	SQR
AND	DELETE	GOTO	MOD	PRINT	STEP
ASC	DIM	HEX	MOTOR	PSET	STICK
AUTO	DRAW	IF	NAME	PUT	STOP
BEEP	EDIT	INKEY	NEW	RANDOMIZE	STRING
CALL	ELSE	INPUT	NEXT	REM	SYSTEM
CHR	END	INT	NOT	RENUM	TAB(
CIRCLE	EOF	KEY	OFF	RESET	TAN
CLEAR	EQUAL	KILL	ON	RESTORE	THEN
CLOSE	ERASE	LET	OPEN	RESUME	TIME$
CLS	ERR	LINE	OPTION	RETURN	TO
COLOR	ERROR	LIST	OR	READ	TROFF
COM	EXP	LLIST	OUT	RIGHT	USING
COMMON	FIELD	LOAD	PAINT	RND	VAL
CONT	FILES	LOCATE	PAINT	RSET	WAIT
COS	FIX	LOG	PEEK	RUN	WEND
CSNG	FOR	LPRINT	PEN	SAVE	WHILE
DATA	FRE	LSET	PLAY	SCREEN	WRITE
DATE	GET	MARGE	POS	SGN	XOR

10.4 TYPES OF ENTRIES IN BASIC

You can make three types of entries in BASIC. These are Commands, Program statements and data. Commands and Program statements can be entered at any time. Data can be entered with the Data or assignment statement or when asked by the computer.

10.5 COMMANDS

Commands are executable instructions such as SAVE, LIST, DELETE, LOAD, NEW, etc. Some BASIC commands are described below:-

AUTO

It generates an automatic line number each time **<ENTER>** key is pressed.

AUTO — When line number and increment are not specified, then both assume the value 10. On pressing **<ENTER>** ↵ key each time a new line number is displayed as 10, 20, 30 … and so on.

AUTO 80, 20 First number specifies the beginning of line number and second number specifies the increment in the line numbers.

AUTO , 20 If line number is not specified before comma (,) then the first line number assumes 0 value.

AUTO ., If period (.) is entered before comma (,) it indicates the current line. If increment is not specified then the increment last specified is assumed. If current line number is 120, and previous increment is 20, then on pressing **<ENTER>**↵ key, the line numbers displayed will be 120, 140, 160 ...and so on.

CLEAR

It clears value assigned to numeric and string variable, resuming all numeric variables to zero and all string variables to null and closes all open files.

CLS / HOME It clears all files and clears all common and user variables. It resets the stack, string spaces and turns off any sound. On some computer it is HOME and on others it is CLS (abbreviation for "**CL**ear **S**creen"). Executing CLS or HOME clears the screen and the cursor returns on the top left hand corner of the cleared screen.

DELETE

It deletes Program lines or line range.

DELETE 50 It deletes the specified line whose line number is given. In this example line # 50 will be erased.

DELETE 30-80 It deletes all the lines within the range given. In this example, lines # 30 to line # 80 will be erased.

DELETE -110 It deletes all lines upto the end line number entered, In this example all the lines upto 110 will be erased.

DELETE 70- It deletes all lines from the line number specified upto the end of the program if end line number is not mentioned.

EDIT

It edits a specified line number.

EDIT 40 The EDIT command displays specified line with the cursor under first digit of line. Changes can be made with editing keys.

FILES

To display names of all files from the specified drive.

FILES "(filespec)" Filespec is a valid file if a filespec is omitted, all files in current directory of the selected drive are displayed.

FILES display all the filenames in current directories

FILES "BAS" all the files having BAS extension.

FILE "A" display all the files on drive A.

KILL

To delete a file from disk

KILL<filename> KILL is used to delete all types of disk files including program, random data and sequential data files. If the line number is followed by comma and increment is not specified, then the last increment specified is assumed.

LIST

To display the program or part of the program currently in memory.

LIST	lists all the program and lines currently in memory.
LIST 20-40	lists all programs between line numbers specified (#20 to #40).
LIST 40 -	lists all programs from line number specified (#40 onward).
LIST - 60	lists all programs upto the end line number specified.

LOAD

To load a file from diskette into memory.

LOAD — The LOAD command retrieves a program that was saved by the SAVE command and read it into the memory. Execution of the LOAD command closes all open files and deletes all variables and program lines currently in the memory before execution.

LOAD "drive name (file name)", R.

If R option is specified with LOAD, the program runs automatically, immediately it is loaded and all open data files are kept open. LOAD with R option allows chaining several programs (or segment of the same program). When file name is omitted, the program file which is found first will be read.

NEW

To delete the program currently in memory and clears all variables.

NEW — It erases the old program from the memory, ensuring that lines from a previous program will not affect the new program. On some computers NEW just clears the memory of the computer but does not clear the screen, it may still be on the screen.

RENUM

To renumber program lines without changing the order of program lines.

RENUM 100, 35, 10 — First number specifies new number in the new sequence for the first line. Second number specifies old number of the line in the current program whose renumbering is to begin. Third number specifies increment to be used in the new sequence.

RENUM — If line numbers are not specified then the first new line number will be 10 with an increment of 10. It also changes all old references in the program by new line numbers.

RUN

It begins program execution currently in memory from the lowest line number or to load it from the diskette into memory and execute it.

RUN 50 — If line number is specified, execution begins from that line.

SAVE

It is used to save a program file on a derive.

SAVE — If filename already exists, the file will be over written.

SAVE "A: TEST" — It saves the file "TEST" on the disk in drive A. If drive name is not mentioned, then it will be saved on the current drive.

SYSTEM

It is used to exit BASIC to return to DOS by entering the SYSTEM command. **Save your program** before pressing <ENTER> ↵ key.

10.6 STATEMENTS

Program statements such as REM, LET PRINT, END etc., are the part of a computer program. They have specific purposes in a computer program. The difference between commands and statements is that **commands** are generally executable in the direct mode. Commands usually perform some type of program maintenance such as editing, clearing, running, or saving programs. On the other hand **statements** are the part of a computer program and run in compiler's mode.

A program statement in BASIC has two parts in general; a line number and one or more BASIC instructions. Line number tells the position of a statement in the program and must be a positive integer from 0 to 65529. For example:-

```
50 LET NUM = 18
```

A statement with a smaller line number (say) 30 will execute before a statement with a larger line number (say) 80, regardless of when it is entered.

10.7 DATA TYPE

BASIC keeps its data types simple. It recognizes numerals and strings:-

NUMERIC DATA

BASIC deals with data having numbers that are used in calculations. For example: 174, 16, and 39.82 etc.

STRING DATA

BASIC also deals with words or texts called strings. A string consists of all the characters words or numbers enclosed by quotation marks such as; "786", "PAKISTAN DAY", "14TH AUGUST 1947" etc. However arithmetic operations cannot be performed on numbers entered with quotation marks.

CONSTANTS

A constant in BASIC is something that does not change its value during execution of the program. There are two types of constants; Numeric constants and string constants.

NUMERIC CONSTANTS

Numeric constants are positive or negative numbers. Plus sign is optional for positive numbers. For examples: 23, -4.756, and +38.15 etc. A numeric constant in BASIC cannot contain commas. Thus 2,45,765 is not a valid number.

STRING CONSTANTS

A string constant is any set of (maximum 255) characters enclosed in quotation marks. Blank spaces may also be included but not quotation marks. Such as "I LOVE PAKISTAN", "23RD MARCH", and "43+17 = ?" etc.

VARIABLES

Variables are the names used to represent certain quantities in a BASIC program. These variable names are usually made up of one or more characters. The number of characters that can be used varies in different versions of BASIC. There are two types of variables; numeric variables and string variables.

NUMERIC VARIABLES

A numeric variable always has a numeric value either supplied by the user or calculated by the computer during program execution. A numeric variable name may consist of all **alphabets** and **numeric constants** but must **begin** with an alphabet. Special characters and blank spaces are not allowed in numeric variable names. Valid variable names which are allowed in BASIC are A, B2, X55, NUM, TPAY etc. Examples of invalid variable names are:-

INVALID VARIABLE NAME	REASON	VALID VARIABLE NAME
5NUM	cannot begin with a digit	NUM5
TOT PAY	Blank spaces are not allowed	TOTPAY
NET-PAY	Hyphen not allowed	NETPAY
B&W	Special symbol (&) not allowed	BANDW
TAX$	$ symbol invalid for numeric variable name.	TAX
LIST	cannot use reserve words.	

STRING VARIABLES

A string variable represents a character string that is a character or a sequence of characters i.e., letters, numbers and special characters such as +, –, *, $, =, ?, etc. Blank spaces may be included in a string but not the quotation marks. A string variable name must be followed by a dollar sign ($) such as A$, B33$, NAM$, GRAD$ etc. A string variable name is valid under similar conditions as for numeric variable except that its last character must be $.

10.8 OPERATORS

Operators perform mathematical or logical operations on values. In BASIC the operators are divided into four categories.

- Arithmetic.
- Relational
- Logical.
- Functional

ARITHMETIC OPERATORS

Numeric expressions contain operators that perform arithmetic operations such as addition, subtraction, multiplication etc. These are called arithmetic operators. The following special symbols are used in BASIC language.

TABLE 10.3 Showing the Arithmetic Operators used in BASIC

OPERATION	OPERATOR	EXAMPLE IN	
		ARITHMETIC	BASIC
Addition	+	5 + 8	5 + 8
Subtraction	–	8 – 3	8 – 3
Multiplication	*	4×7	4 * 7
Division	/	$15 \div 5$	15 / 5
Exponentiation	^	9^2	9 ^ 2
Negation	–	– 6	– 6

10.9 THE HIERARCHY OF ARITHMETIC OPERATORS

The computers follow the usual rules of arithmetic for evaluation of expressions. The order in which various operations are performed is as follows:-

First step:	Perform all operations inside parentheses.
Second step:	Evaluate all exponentiation from right to left.
Third step:	Perform all multiplications and divisions from left to right.
Fourth step:	Perform all additions and subtractions from left to right.

BASIC ARITHMETIC EXPRESSIONS

An expression is a set of mathematically meaningful combination of variables, constants and operators. Equations and formulae are also expressions. Arithmetic operations are performed using the operators described in Table 10.3.

Arithmetic expressions in BASIC are written on a single line which may cause confusions and would produce wrong results. For example computer will interpret an expression $\frac{X+Y}{2}$ written on a single line as X+Y/2 that is incorrect. To avoid confusion brackets or parentheses may be added to write expressions in a single line. Thus the above expression can be written correctly on a single line as (X+Y)/2. Let us study some more examples.

EXAMPLE 10. 1

Evaluate the expression 5 + 2 * 3.

SOLUTION: $5 + 2 * 3 = 5 + 6$ (Performing Multiplication first)

$= 11$

EXAMPLE 10.2

Evaluate the expression $(5 + 2) * 3$.

SOLUTION: $(5 + 2) * 3 = 7 * 3$ (Opening the parentheses first)
$= 21$

EXAMPLE 10.5

Evaluate the expression $\sqrt{B^2 - 4AC}$, where A = 2, B = 5, and C = – 3.

SOLUTION:
$$
\begin{aligned}
\sqrt{B^2 - 4AC} &= (B \wedge 2 - 4 * A * C) \wedge 0.5 \\
&= \{5 \wedge 2 - 4 * 2 * (-3)\} \wedge 0.5 \\
&= (25 + 24) \wedge 0.5 \\
&= 49 \wedge 0.5 \\
&= 7
\end{aligned}
$$

RELATIONAL OPERATORS

Relational operators are used to relate or compare two quantities. The result of comparison may be either YES or NO or may be TRUE or FALSE. When these operators appear in a computer program, their results decide the flow of the computer program. The Table 10.4 below shows symbols used for relational operators, their relations with examples.

TABBLE: 10.4 Relational Operators, their relations with illustrations.

OPERATOR	RELATION	EXAMPLE
=	Equality	If A = 15 Then STOP.
$<>$	Inequality	If D$ $<>$ SUNDAY Then display **WELCOME.**
$<$	Less than	If M $<$ 50 Then Print FAIL.
$>$	Greater than	If M $>$ 50 Then Print CONGRATULATION.
$<=$	Less than or Equal to	If K $<=$ 10 Then Go to Step 70.
$>=$	Greater than or equal to	If G $>=$ 17 Then X = X*2 Otherwise X = X*1.5

It should be noted that Arithmetic Operators are performed first when they appear along with the Relational Operators in the same expression.

LOGICAL OPERATORS

Logical Operators such as AND, OR, NOT, NAND, NOR, and XOR perform in the same way as in Boolean algebra. The result of these logical operations would be either true or false. In a Computer program, such results are very important because they decide the flow of a program.

For example an allowance of Rs.1000 is announced for employees under grade 12 of firms whose staff members are more than 4. It may be expressed as:-

IF G < 12 AND MEM > 4 THEN ALL = 1000

Excellent grade is awarded to students getting 80 % or more marks or total marks 360 or above. This can be expressed as:-

IF M >= 80 OR TOT >= 360 THEN GR$ = EXCELLENT

FUNCTIONAL OPERATORS

BASIC is very rich in having predetermined functions that perform arithmetic and string operations. These are built-in functions such as ABS, INT, FIX, RND, SQR, LOG, SIN and COS. These functions make the programming much simpler. We shall discuss them in detail in Chapter 13.

10.10 MODES OF OPERATIONS

Once BASIC is loaded, it displays the **Ok** prompt. Ok means BASIC is at command level, ready to accept BASIC commands Program statements can be entered with or without a line number, depending on the mode of operation: **direct mode** or the **program mode**. For Labs. consult **Computer Lab. Journal** for SSC. Computer Science by Prof. M. Tahir Hassan.

DIRECT MODE

In the **direct mode,** program statements are entered without a line number. The computer will execute it as soon as the <RETURN> or <ENTER> ↵ key is pressed after typing the Command. Output is displayed on the screen. This facility can turn BASIC into a calculator to provide immediate computation.

10.11 THE PRINT COMMAND

One of the most useful words in BASIC is PRINT. The PRINT command allows the computer to communicate with you. It can be used to display any combination of letters, numbers, and symbols, or to evaluate numerical expressions. When strings are used with the PRINT command, they must be enclosed in quotation marks, as in the following examples:

PROGRAM 10.1

```
PRINT "IN THE NAME OF ALLAH THE MOST BENEFICIENT AND THE MOST MERCIFUL"
```

Press <ENTER> ↵ key after typing this Example 10.1. The output will be:

OUTPUT OF PROGRAM 10.1

```
IN THE NAME OF ALLAH THE MOST BENEFICIENT AND THE MOST MERCIFUL
```

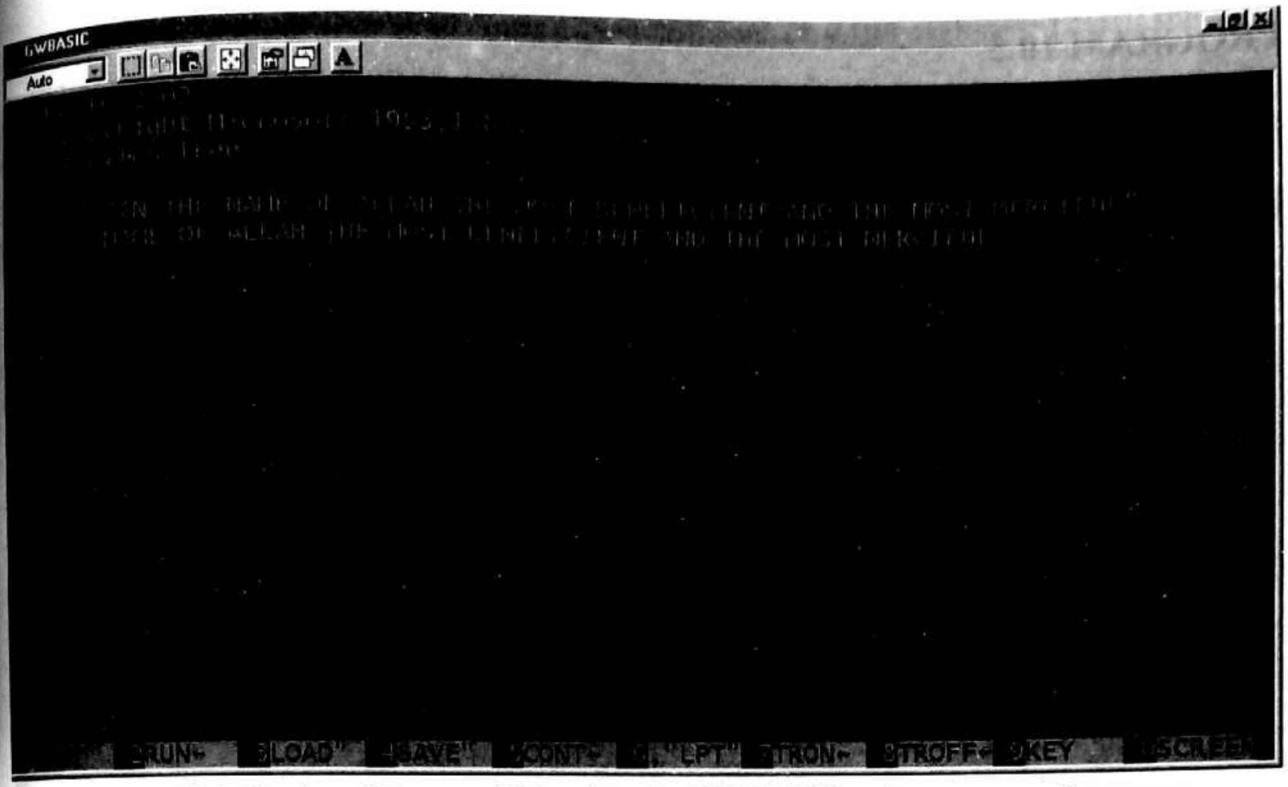


Figure 10.1 Display of Program10.1 written in GW-BASIC as it appears on the screen.

Let us have more examples using PRINT command.

PROGRAM 10.2

```
PRINT  25 + 39
```

Press <**ENTER**> ↵ key after typing Example 10.3. The output will be:

OUTPUT OF PROGRAM 10.2

```
64
```

PROGRAM 10.3

```
PRINT  " 25 + 39 "
```

Press <**ENTER**> ↵ key after typing Example 10.3. The output will be:

OUTPUT OF PROGRAM 10.3

```
25 + 39
```

PROGRAM 10.4

```
PRINT  " 25 + 39 =  " 25 + 39
```

Press <**ENTER**> ↵ key after typing Example 10.4. The output will be:

OUTPUT OF PROGRAM 10.4

```
25 + 39 =  64
```

PROGRAM 10.5

```
PRINT  " THE SUM OF 25 AND 39 IS   " 25 + 39
```

Press <**ENTER**> ↵ key after typing Example 10.5. The output will be:

OUTPUT OF PROGRAM 10.5

```
THE SUM OF 25 AND 39 IS  64
```

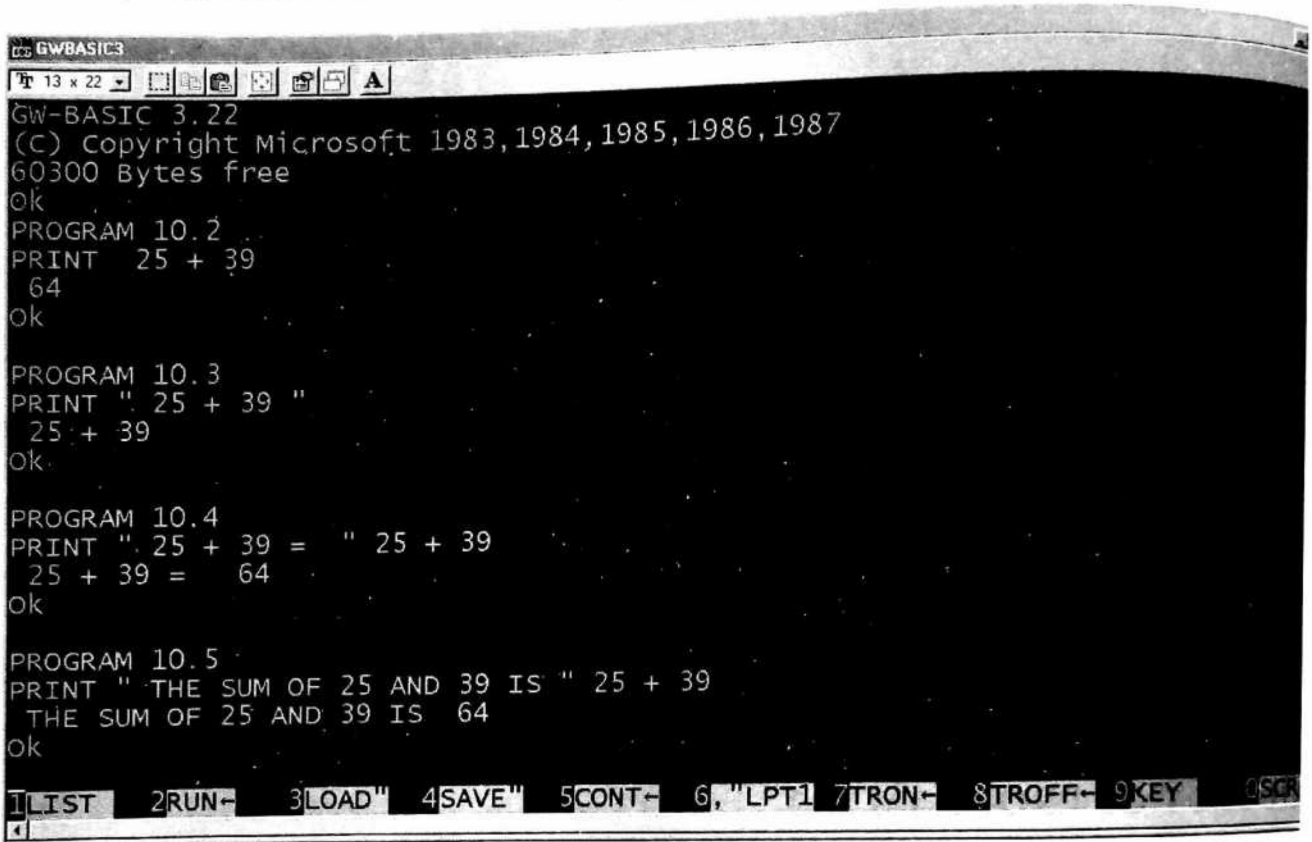


Figure 10.2 Displays Programs 10.2, 10.3, 10.4 and 10.5 written and executed successively in GW-BASIC as they appear on the screen. Note the effect of quotation marks.

PROGRAM 10.6

You can use ALT + P as a shortcut for typing PRINT ; or type question mark ? to execute the PRINT command as illustrated in the following program.

```
? ( 28 + 36 )/5
```

Press <**ENTER**> ↵ key after typing Example 10.4. The output will be:

OUTPUT OF PROGRAM 10.6

```
12.8
```

PROGRAM MODE

The program mode is also called the indirect mode. A program in BASIC consists of line number followed by an instruction or statement. The statement

with the smallest number is executed first followed by the statement with the next smallest number and so on.

10.12 STRUCTURE OF BASIC PROGRAM

A BASIC program is composed of statements which execute in the order in which they are numbered unless otherwise indicated by the program. General structure of the BASIC program is as under:-

- Every statement must appear on a separate line.
- Two statements typed in the same line must be separated by a colon.
- Every Program statement is entered with line number that is used to determine the sequence of its execution.
- No two program statements have the same line number.
- Each line number must be followed by BASIC keyword that indicates the type of instruction to be carried out.
- Blank spaces may be inserted wherever desired to improve the readability of program statement.

A Structured Approach to BASIC

In its early forms, BASIC was often criticized as being unstructured, because of the use of GOTO instruction and thus create inefficient programs. Yet it is very simples to write structured programs in GW-BASIC. In a structured approach we use three types of building blocks, the processing block, the decision block, and the repetition block.

10.13 THE PROCESSING BLOCK

The **processing block** mainly consists of **input statements** that allow data to be entered, **assignment statements** that allow calculation of the data and **output statements** that display the results on the screen or print them. Let us begin with the following BASIC's statements such as PRINT, LET, and STOP/END statements. These statements are of elementary nature.

PRINT STATEMENT

PRINT statement when executed, displays the processing results of the computer. You are already familiar with the PRINT command when used in the direct or interpreter mode. The only difference is that in the program or compiler mode, PRINT is followed by a line number such as:-

Line # **PRINT** {
Variables (A, TAX, NAM$)
Label (string constant enclosed within quotation marks)
Arithmetic expression (37 - 3 ^ 2 * 4 + 15)
Any combination of the above

Let us write programs 10.7A 10.7B using PRINT , END, and STOP statements to get the average of two numbers as follows:-

PROGRAM 10.7A

```
10 CLS
20 ? " Average = "; ( 40 + 23 ) / 2
30 END
```

PROGRAM 10.7B

```
10 CLS
20 ? " Average = "; ( 40 + 23 ) / 2
30 STOP
```

On pressing the <**ENTER**> ↵ key, program will be stored in the memory. Now type RUN command without any line number and press <**ENTER**> ↵ key or press key F2 for execution. The result and the program will appear as:

DISPLAY OF PROGRAM 10.7A

```
10 CLS
20 ? " Average =" ; ( 40 + 23 ) / 2
30 END
RUN
 Average = 31.5
Ok_
```

DISPLAY OF PROGRAM 10.7B

```
10 CLS
20 ? " Average =" ; ( 40 + 23 ) / 2
30 END
RUN
 Average = 31.5
Break in 20
Ok_
```

10.14 LET STATEMENT

LET is an assignment statement that stores a value in a location assigned to a given variable. The general format of the LET statement is as follows:-

Line # LET variable = expression

Let us write a program using LET statement to calculate the average of two numbers.

PROGRAM 10.8

```
10 LET A = 5
20 LET B = 11
30 LET C = ( A + B ) / 2
40 PRINT C
50 END
```

OUTPUT OF PROGRAM 10.8

```
RUN
 8
Ok_
```

Let us write a program to print " I LOVE PAKISTAN " as follows:-

PROGRAM 10.9

```
10    LET    A$ = "I"
20    LET    B$ = "LOVE"
30    LET    C$ = "PAKISTAN"
40    PRINT  A$, b$, C$
50    END
```

Type RUN and press <ENTER>↵ key, the output appears as:-

OUTPUT OF PROGRAM 10.9

```
I     LOVE     PAKISTAN
```

10.15 COMMAS (,) AND SEMICOLONS (;)

Commas and semi colons have a special purpose in BASIC. When used in PRINT statement, they control the position of output on the screen.

THE COMMA

A comma in PRINT statement separates strings and numerical expression. It generally causes the output to be printed across the screen in five zones.

PROGRAM 10.10

```
10 ? "TODAY", "ROLL", "NOS" 122, 54, 136 "ARE", "ABSENT "
20 END
```

OUTPUT OF PROGRAM 10.10

```
TODAY     ROLL     NOS      122      54
136       ARE      ABSENT
```

When there will be more numbers or strings than zones in a line, then the computer would automatically PRINT those extra numbers or strings beginning from the first zone of the next line.

THE SEMICOLON

Semicolon in PRINT statement also affects the form of output in a different ways. The computer prints the strings without adding space when semi-colon are used in a PRINT statement to separate strings. However in numerical expression PRINT statement varies from computer to computer. For example let us see the output for the above statement with semicolons instead of commas.

PROGRAM 10.11

```
100 ? "TODAY";"ROLL";"NOS";122 ; 54 ; 136 ; "ARE" ; "ABSENT"
200 END
```

OUTPUT OF PROGRAM 10.11

```
TODAYROLLNOS 122 54 136 AREABSENT
```

10.16 ASSIGNMENT STATEMENTS

The assignment statement is used when the programmer wishes to give a variable an initial value that will be used throughout the program, or when a variable is assigned a new value as a result of computation or operation. For example the following statements assign number " 5 " to the variable M, and name " KASHIF NAEEM " to the variable NAM$:-

```
40   M = 34
90   NAM$ = " KASHIF NAEEM "
```

The assignment statement has the following format:-

Line # (LET) variable name = variable or expression

The word LET is optional in BASIC assignment. The term expression is defined as a combination of two or more variables and/or one or more constants and operators. The following expressions are valid:-

63, K + 2 and 5 * X ^ 2 – 3 * Y etc.

Table 10.4 contains examples of valid assignment statements and their results. In the last example, the value of a variable N is increased by one and this new value replaces the previous value. The following assignment statement is not a valid statement.

A + B = X

Table 10. 4 Examples of valid Assignment Statements

Assignment Statement	Result
P = Q	Current value of Q is placed in P
NUM = K + T	Current values of K and T are summed and placed in NUM
K = 0	K is set to 0
C$ = "IRAN, TURKEY "	The string variable C$ is set equal to "IRAN, TURKEY "
U$ = C$ + "AND PAKISTAN"	Stores "IRAN, TURKEY AND PAKISTAN" in the string variable U$
N = N + 1	Old value of N is increased by one to form a new value of N

Program 10.12 gives product of two numbers using assignment statement.

PROGRAM 10.12

```
10   P = 25
20   Q = 45
30   PRINT P * Q
40   END
```

OUTPUT OF PROGRAM 10.12

```
1125
```

Program 10.13 prints " IRAN, TURKEY AND PAKISTAN " using assignment statements as given below:-

PROGRAM 10.13

```
10  C$ = " IRAN, TURKEY "
20  U$ =  C$ + " AND PAKISTAN "
30      PRINT U$
40      END
```

OUTPUT OF PROGRAM 10.13

IRAN, TURKEY AND PAKISTAN

10.17 REMark STATEMENT

The REM (remark) statement documents the program and provides information to the user about the program. It may explain what a program is? REM statements state the purpose of the program, necessary instructions for its execution and its logic to users. REM statements are non-executable and do not instruct the computer in any way. Thus it can be placed anywhere in the program.

PROGRAM 10.14

```
10  REM * THIS PROGRAM COMPUTS EMPLOYEE'S NET PAY *
10  LET PAY = 3450
20  LET HRENT = 1035
30  LET DED = 417
40  LET NETPAY = PAY + HRENT - DED
50  ? NETPAY
60  END
```

OUTPUT OF PROGRAM 10.14

4068

10.18 THE INPUT STATEMENT

So far you have learnt a method of assigning values to variables using the LET statement. BASIC provides another way of assigning values to variables using the INPUT statement. This statement allows you to enter value of a variable while the program is running. The format of the INPUT statement is:

Line # INPUT (Variable name)

Let us write a program to calculate the sum of two numbers.

PROGRAM 10.15

```
10 INPUT A
20 INPUT B
30 C     = A + B
40 PRINT C
50 END
```

Type RUN and press <ENTER> ↵ key. Question mark " ? " will appear asking you for the value of A. Enter a value for A. Now the prompt " ? " will appear at the next line asking you for the value of B. Enter a value for B. The computer will execute as:-

OUTPUT OF PROGRAM 10.15

```
? 37
? 44
 81
```

To display the message "SUM OF A (*value*) AND B (*value*) IS C (*value*), the above Program 10.15 can be modified as:-

PROGRAM 10.16

```
 5  CLS
10  INPUT  A
20  INPUT  B
30  C       = A + B
40  ? " SUM OF  "; A; " AND  ";  B;  "  IS  "; C
50  END
```

Type RUN and press <ENTER> ↵ key. Enter the values for A and B after the ? prompt. The output will be:-

OUTPUT OF PROGRAM 10.16

```
? 487
? 218
 SUM OF 487 AND 218 IS 705
```

A single INPUT statement may be used to assign values of more than one variable separated by commas. The program 10.16 can be modified as follows:-

PROGRAM 10.17

```
10  CLS
20  INPUT    A, B
30  C       =   A+B
40  ? "   SUM OF   "; A; " AND ";    B; "  IS "; C
40  END
```

Type RUN and press <ENTER> ↵ key. Enter the values for A and B separated by comma (as in the INPUT statement) after the ? prompt. The output of this program will be similar to that of Program 10.16 as follows:-

OUTPUT OF PROGRAM 10.17

```
? 35, 29
 SUM OF 35 AND 29 IS 64
```

As line 20 is executed, the ? prompt appears on the screen. In this case enter two values (say) 35 and 29 separated by comma for the variables A and B respectively. The computer will accept both these values as soon as you enter them.

In above programs the computer will wait for values after ? prompt. If a Prompt (*a string constant*) is included, then a meaningful message may be displayed before the question mark. The format of INPUT statement will then be:-

line # INPUT " PROMPT "; (variable), (variable) …

PROGRAM 10.18

```
10  CLS
20  INPUT   " PLEASE ENTER A, B  " ; A, B
```

```
30  PRINT  " SUM OF  " ; A ; "  AND  " ; B ; "  IS " ; A + B
40  END
```

OUTPUT PROGRAM 10.18

```
PLEASE ENTER A, B? 27, 41
SUM OF  27  AND  41 IS  68
```

" PLEASE ENTER A, B? " appearing on execution may be made without the prompt " ? " replacing semicolon by a comma after the quotation mark in line 20 of Program 10.18 as 20 INPUT " PLEASE ENTER A, B ", A, B
Note the difference in prompts using comma and semicolon in Program 10.19.

PROGRAM 10.19

```
10  CLS
20  INPUT    " PLEASE ENTER A " , A
30  INPUT    " PLEASE GIVE B " ; B
40  PRINT  "  SUM OF  " ; A ;  "  AND  " ; B ; "  IS "  ; A + B
50  END
```

OUTPUT PROGRAM 10.19

```
PLEASE ENTER A 18
PLEASE GIVE B? 15
SUM OF 18 AND 15 IS 33
```

PROGRAM 10.20

```
10  INPUT     "  PLEASE ENTER PLAYER'S  NAME :  " ; N$
20  INPUT     " PLEASE ENTER HIS SCORE : ", R
30  PRINT     N$ ;  R ; "  RUNS "
40  END
```

OUTPUT PROGRAM 10.20

```
PLEASE ENTER PLAYER'S NAME : ? SAEED ANWAR
PLEASE ENTER HIS SCORE : 62
SAEED ANWAR 62 RUNS
```

10.19 READ . . . . DATA Statements

So far you have learnt two ways to assign values to variables using LET and INPUT statements. The LET Statement assigns values from within a program, and the INPUT Statement allows you to supply the values while the program is RUNing. There is a third method using READ and DATA statements. This is useful when the data contains large numbers of values. A DATA statement contains values of the variables which are assigned to the respective variable names in READ statement. The format of READ and DATA statements are:-

Line # READ (1st variable name), (2nd variable name), …
Line # DATA (value of 1st variable), (value of 2nd variable), …

READ . . . . DATA techniques are illustrated in the following programs.

PROGRAM 10.21

```
10  READ    A, B, C, D
20  DATA    35, 92
30  DATA    78, 49, 66,
40  PRINT   A ; B ; D ; A ; C ; B
50  END
```

OUTPUT PROGRAM 10.21

```
35 92  49 35 78 92
```

The READ statement may contain numerical variables for the numeric DATA and string variables for the string constants in the DATA statement as illustrated in the following example.

PROGRAM 10.22

```
10      READ     A, B, C, A$, B$, C$
20      DATA    58, 73, 65,
30      PRINT   A$ ; " OBTAINED  " ; A ; "  MARKS "
40      PRINT   B$ ; " OBTAINED  " ; B ;  "  MARKS "
50      PRINT   C$ ; " OBTAINED  " ; C ;  "  MARKS "
60      DATA     BASHIR, ARIF, MUSTANSAR
70      END
```

OUTPUT PROGRAM 10.22

```
BASHIR OBTAINED  58  MARKS
ARIF OBTAINED  73  MARKS
MUSTANSAR OBTAINED  65  MARKS
```

10.20 TAB FUNCTION

BASIC allows lining up the numbers vertically. Let us modify program 10.22 using TAB function. TAB function must be accompanied by a number enclosed by parentheses telling the computer to move cursor to column specified in the parentheses such as TAB(44) moves the cursor to column 44. **Note that any gap between TAB and the left parenthesis will cause Syntax error.**

PROGRAM 10.23

```
10      CLS
10      READ     A, B, C, A$, B$, C$
20      DATA    58, 73, 65,
30      PRINT   A$ TAB(20) " OBTAINED  " ; A ; "  MARKS "
40      PRINT   B$ TAB(20) " OBTAINED  " ; B ;  "  MARKS "
50      PRINT   C$ TAB(20) " OBTAINED  " ; C ;  "  MARKS "
60      DATA     BASHIR, ARIF, MUSTANSAR
70      END
```

OUTPUT PROGRAM 10.23

```
BASHIR                      OBTAINED  58  MARKS
ARIF                        OBTAINED  73  MARKS
MUSTANSAR                   OBTAINED  65  MARKS
```

10.21 PRINT USING STATEMENT

Tab function cannot line up the decimal points correctly. This problem can be solved by PRINT USING statement. PRINT USING statement allows specifying the appearance and position of numbers. It also provides a way to round off numbers to a given number of decimal places. Below is one of the many formats of PRINT USING statement that tells the computer to print a value upto five digits before the decimal point and two digits after the decimal point.

Line # PRINT USING "#####.##"

PROGRAM 10.24

```
10  CLS
20  REM * PRINTING DATA VALUES IN VERTICAL FORMAT *
30  READ A, B, C, D,
40  DATA 304.567, 1145, 97.479, 2.88
50  PRINT USING "####.##" ; A
60  PRINT USING "####.##" ; B
70  PRINT USING "####.##" ; C
80  PRINT USING "####.##" ; D
90  END
```

OUTPUT OF PROGRAM 10.24

```
 304.57
1145.00
  97.48
   2.88
```

Let us develop a program 10.25 that causes text to appear in its output.

PROGRAM 10.25

```
10  CLS
20  REM "RAINFALL IN CM DURING JULY"
30  READ 'A', 'B', C, D, 'A$, 'B$, C$, D$
40  DATA 4.85, 13.2, 11.68, 1.6, FIRST WEEK
50  PRINT A$ ; TAB(20) ;
55  PRINT USING "##.##" A
60  PRINT B$ ; TAB(20) ;
65  PRINT USING "##.##" B
70  PRINT C$ ; TAB(20) ;
75  PRINT USING "##.##" C
80  PRINT D$ ; TAB(20) ;
85  PRINT USING "##.##" D
90  DATA SECOND WEEK, THIRD WEEK, FOURTH WEEK
100 END
```

OUTPUT OF PROGRAM 10.25

```
FIRST WEEK            4.85
SECOND WEEK          13.20
THIRD WEEK           11.68
FOURTH WEEK           1.60
```

SUMMARY

BASIC PROGRAMMING

BASIC is the first high level languages to be used on micro-computers. First interpreter of BASIC language was written in 1975 by Microsoft.

CHARACTER SET IN BASIC LANGUAGE

Character set in BASIC language consists of alphabetic characters of English language, numeral characters, special characters, arithmetic, relational, logical and functional operators.

Reserved Words

Certain words have special meaning in BASIC. These words are called reserved words such as AND, AUTO, CLS, DIM, FOR, LET, NEXT STEP etc.

COMMANDS

Commands are executable instructions such as SAVE, LIST, LOAD, etc.

STATEMENTS

Program statements such as REM, LET PRINT, END etc., are the part of a computer program. The difference between commands and statements is that **commands** are generally executable in the direct mode. Commands usually perform some type of program. **Statements** run in compiler's mode.

DATA TYPE

BASIC has data types consisting of numeric data, string data.

CONSTANTS

There are two types of constants; Numeric constant and string constants.

VARIABLES

There are two types of variables; numeric variables and string variables.

Numeric Variables

A numeric variable name may consist of all **alphabets** and **numeric constants** but must **begin** with an alphabet. Valid variable names which are allowed in BASIC are A, B2, X55, NUM, TPAY etc.

String Variables

A string variable represents a character string such as letters, numbers and special characters such as +, –, *, =, $, ?, etc. including blank spaces but not the quotation marks. A string variable name must be followed by a dollar sign ($).

OPERATORS

Operators perform mathematical or logical operations on values. In BASIC there are Arithmetic, Logical, Relational, Functional operators.

The Hierarchy Of Arithmetic Operators

First step:	Perform all operations inside parentheses.
Second step:	Evaluate all exponentiation from right to left.
Third step:	Perform all multiplications and divisions from left to right.
Fourth step:	Perform all additions and subtractions from left to right.

Modes Of Operations

Once BASIC is loaded, it displays the **Ok** prompt. Ok means BASIC is at command level, ready to accept BASIC commands. Program statements can be entered in **direct mode** or in the **program mode**.

BASIC PROGRAM

A BASIC program is composed of statements which execute in the order in which they are numbered unless otherwise indicated by the program.

PRINT STATEMENT

PRINT statement is similar to PRINT command used in compiler mode.

LET STATEMENT

LET is an assignment statement that stores a value in a location assigned to a given variable. The word LET is optional in BASIC assignment.

COMMAS (,) AND SEMICOLONS (;)

Commas and semi colons have a special purpose in BASIC. When used in PRINT statement, they control the position of output on the screen.

ASSIGNMENT STATEMENTS

The assignment statement is used when the programmer wishes to give a variable an initial value that will be used throughout the program, or when a variable is assigned a new value as a result of computation or operation.

REMark STATEMENT

The REM (remark) statement documents the program and provides information to the user about the program. REM statements are non-executable.

THE INPUT STATEMENT

This allows entering value of a variable while the program is running.

READ . . . . DATA STATEMENTS

DATA statement contains values of variables present in READ statement. READ statement assigns those values to the respective variables in the list.

PRINT USING STATEMENT

PRINT USING statement allows specifying the appearance and position of numbers. It also rounds off numbers to a given number of decimal places.

EXERCISE

10.01 Complete the following statements

i) BASIC stands for ______________________.

ii) Basic is well suited for computers having ____________ memory.

iii) In BASIC key F1 is equivalent to the command ____________.

iv) __________are the_ part of program which run in compiler's mode.

v) To run a program press __________ key.

vi) ________________ statements allow you to enter value of a variable while the program is running.

10.02 Tick ( ) the following statements either True or False.

i) Commands but not statements can be entered at any time. True/False

ii) If a filename already exists, then SAVE will over write it. True/False

iii) A string constant cannot contain quotation marks. True/False

iv) REM statements are executable statements . True/False

v) A computer prints strings without adding spaces when strings are separated by semicolon in print statement. True/False

vi) PRINT sends output to a printer. True/False

vii) INPUT is an executable statement. True/False

viii) DATA statement may be placed anywhere in a program. True/False

10.03 Encircle one choice A, B, C or D in each case.

i) Which of the following word can be used as a variable name?
(A) COLOR (B) KEY (C) PEN (D) PENCIL

ii) Which of the following command is used to exit from BASIC and to return to DOS?
(A) CLEAR (B) KILL (C) SYSTEM (D) NEW

iii) The maximum characters that a string constant may contain are :-
(A) 8 (B) 32 (C) 80 (D) 255

iv) Which of the following is not a logical operator?
(A) AND (B) OR (C) FIX (D) NOT

v) Which of the following types of the characters can be used as numeric variable name?
(A) special (B) blank spaces (C) hyphen (D) none of them

vi) Equality operator is a:-
(A) Logical operator
(B) Funtional operator
(C) Relational operator
(D) Arithmetical operator

10.04 Match the items given in Column I with those in Column II

Column I		Column II	
i)	Functional operator	a)	F1
ii)	Save	b)	AND
iii)	Command	c)	F2
iv)	Load	d)	RND
v)	Logical operator	e)	F4
vi)	List	f)	LESS THAN
		g)	PRINT
		h)	F3

10.05. Write a short note on BASIC and describe some of its important features.

10.06 What type of entries are made in BASIC? What is difference between commands and statements?

10.07 Describe briefly the following commands:-
Auto, Delete, List, Renum and System.

10.08 How many types of variable names are used in BASIC? Give examples of invalid variable names with explanations.

10.09 Describe various types of constants used in BASIC with examples.

10.10 Pick out numeric and strings from the following data items:-
"LAHORE" ; 27; "60W"; 1.008; "16"; 20 km ; "23RD MARCH"; IX;A-1.

10.11 Explain various types of operators used in BASIC with examples?

10.12 How does computer know whether an instruction you typed is a program?

10.13 What commands clears the computer's memory?

10.14 What will be the output? When you enter PRINT " 37 C = " 37 * 9 / 5 + 32

10.15 Write a note on assignment statements with examples.

10.16 What is the difference between STOP and END Statement? Explain it with examples.

10.17 Write a program that prints three numbers, their sum and average using LET statement.

10.18 Write a program that prints three numbers, their sum and average using INPUT statement.

10.19 Write a program for the square and cube of any number using INPUT statement. Print the number, its square and cube in zones.

10.20 Tell whether the following are valid BASIC statement/program. If not, give a possible correction and its output if any.

a)
```
10  INPUT "  ENTER A NUMBER  "; N
```
b)
```
10  INPUT "     NUMBERS =  " X, Y, Z
```
c)
```
10  INPUT, A $ + 2
```
d)
```
10  READ A; B
20  PRINT  6, 9
30  PRINT  A + B
```
e)
```
10  READ A$, B $; C $
20  PRINT "   SUBJECTS    "
30  DATA ENGLISH, PHYSICS, MATHEMATICS
40  END.
```
f)
```
10  READ A, B, C
20  LET X = (A + B)* C
30  PRINT X
40  END
```
g)
```
10  INPUT A, B$
20  C = A + B$
30  PRINT A: PRINT B$: PRINT C
40  END
```

10.21 What will be the output of the following program?

```
10   READ A$
20   PRINT A$
30   RESTORE
40   READ B$
50   PRINT B$
60   RESTORE
70   READ C$
80   PRINT C$
90   DATA  ASLAM, AKRAM, AFZAL
100  END
```

10.22 Write a BASIC program to calculate the sum, product and average of 4 numbers using INPUT statement and READ – DATA statements.

11

CONTROL STATEMENTS

In this Chapter, decision and repetition blocks will be discussed. These two blocks are called the **Control structures**. They control the flow of statements execution within the program. The **decision block** controls the statements to be executed, while the **repetition block** controls the repeated execution of one or more statements.

11.1 The GOTO Statement

In all the programs discussed so far, the computer executed the instructions in numerical order, from the smallest line number to the largest one until it reached the last statement. However the order in which program statements are executed can be changed using a control statement. One of the control statements is the GOTO statement that causes the control to transfer anywhere within the Program. This is illustrated in the following programs.

PROGRAM 11.1

```
10    REM * ENDLESS GOTO LOOP OF WELCOME MESSAGE*
20    INPUT " PLEASE ENTER YOUR NAME:" ; A$
30    PRINT " WELCOME" ; A$
40    GOTO 30
50    END
```

OUTPUT OF PROGRAM 11.1

```
PLEASE ENTER YOUR NAME?
```

The computer is asking for the data value of A$ as required by INPUT statement in line # 20. Type anything after the question mark [?] and then press **<ENTER>** ↵ key. The output may be as given below:-

OUTPUT OF PROGRAM 11.1

```
PLEASE ENTER YOUR NAME ? NASIR
WELCOME NASIR
WELCOME NASIR
•
•
•
```

The computer prints the value entered for A$ after the string in line # 30 and arrives at line # 40. The statement GOTO in line # 40 causes the control to go

to line # 30. It prints the same message a second time and as it reaches line # 40, it is again sent back to line # 30 by the GOTO statement at line # 40. The repetition of printing the message will continue until you break the program.

Line 20 and 30 which the computer executes repeatedly; make up what is called a loop. This is shown schematically in Figure 11.1 of an endless loop.

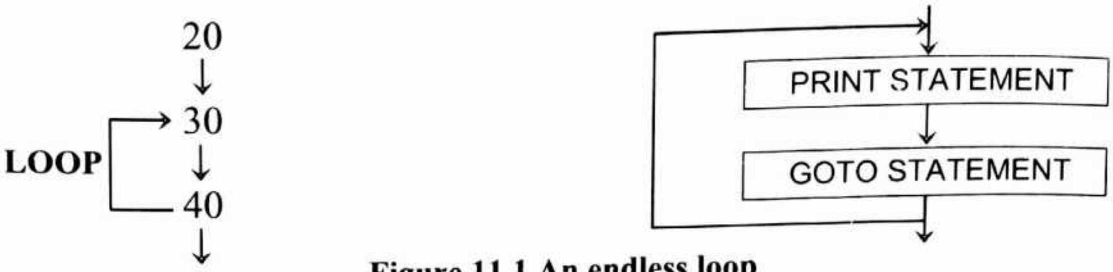


Figure 11.1 An endless loop

As you study the output of program 11.1, the computer continues to PRINT the message WELCOME NASIR over and over again. This is an endless loop. To stop execution of the program, press <BREAK> key (some computers need to press the <BREAK> while holding down the control <CTRL> key). Let us consider program 11.2, which is slightly different than program 11.1.

PROGRAM 11.2

```
10    REM ** ENDLESS GOTO LOOP **
15    REM ** PRINTING OF NUMBERS  FROM 1 TO INFINITY**
20    CLS
30    NUM 1
40    ? NUM ;
50    NUM = NUM +1
60    GOTO 40
70    END
```

As you run the program 11.2, it will start printing the numbers starting from 1, the value of the variable NUM initialized in line # 30 with an increment 1 as instructed by the statement in line # 50 indefinitely until you break the program. The semicolon [;] at the end of line # 40 instructs printing next number with a separation of two spaces after the previous number. If semicolon [;] is replaced by a comma [,] then it will cause numbers to be printed in (5)zones.

OUTPUT OF PROGRAM 11.2

```
:
:
1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790
1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803
1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816
:
:
```

Program 11.1 can be modified to run only once each time the value for A$ is entered by simply transferring the control to line # 20 by the GOTO statement in line # 40. Line # 25 leave a blank line before printing the welcome message. Run program 11.3 with and without line 25 and note the difference.

PROGRAM 11.3

```
10   REM * GOTO LOOP OF WELCOME MESSAGE *
15   REM * RUNS EACH TIME THE VALUE OF A$ IS ENTERED*
20   INPUT "PLEASE ENTER YOUR NAME"; A$
30   PRINT "WELCOME" ; A$
40   GOTO 20
50   END
25   PRINT
```

OUTPUT OF PROGRAM 11.3

```
PLEASE ENTER YOUR NAME ? ASIF
WELCOME ASIF

PLEASE ENTER YOUR NAME ? SALEEM
WELCOME SALEEM

PLEASE ENTER YOUR NAME ? NAVEED
WELCOME NAVEED
.
.
.
```

See the difference in the GOTO statements of program 11.1 and 11.3. In line # 40 of program 11.3 control is sent back to line # 20 instead of line # 30 as in program 11.1, where it is instructed to INPUT another name. The new name (or may be the same name, which is typed the second time) is then PRINTED with the message in line 20. Line 25 causes the computer to leave a blank line in the output. The GOTO statement in line # 40 sends the computer back to line # 20.

11.2 GOTO with READ ... DATA

Loops are particularly useful in programs containing READ.....DATA statements as illustrated in program 11.4.

PROGRAM 11.4

```
10   PRINT "MARKS LIST:"
15   PRINT
20   PRINT "ROLL No." , "NAME", "MARKS"
30   READ R, N$, M
40   PRINT R, N$, M
50   GOTO 30
60   DATA 52408, AFZAL, 742, 63142, SALEEM
70   DATA 695, 47783, RAFIQ, 701
80   END
```

OUTPUT OF PROGRAM 11.4

```
MARK LIST:
ROLL No.        NAME        MARKS
52408           AFZAL       742
63142           SALEEM      695
47783           RAFIQ       701
OUT OF DATA IN 30
```

NOTE The loop in line 30, 40 and 50 allows us to assign three different sets of variables R, N$, M with only one READ statement which is executed several times. The first time line 30 is executed, the READ statement assigns 52408 to R, AFZAL to N $ and 742 to M. The next time line 30 is executed (the second time through the loop), the computer READs the DATA values from the place where it left off previously. This means that 63142 is assigned to R, SALEEM is assigned to N$ and 695 is assigned to M. The third time through the loop, the READ statement assigns 47783 to R, RAFIQ to N$ and 701 to M. After the last values are PRINTED (line 40), the GOTO statement in line 50 sends the computer back to READ statement in line 30. As there is no more value in the DATA list, the computer prints the message OUT OF DATA IN 30 and the program stop.

11.3 GOTO using LET statement

By using GOTO statement and loops, the computer performs tasks repeatedly, quickly and efficiently. Let us write a program for a table of numbers.

PROGRAM 11.5

```
10    REM * ENDLESS GOTO LOOP FOR A TABLE OF NUMBERS *
20    INPUT "  TABLE OF   : " ; S
30    LET N = 1
40    LET P = S *N
50    PRINT S, N, P
60    LET N = N + 1
70    GOTO 40
80    END
```

OUTPUT OF PROGRAM 11.5

```
TABLE OF : ? 8              (Type a number say 8 after?)
     8               1                 8
     8               2                16
     8               3                24
```

In line 30 the value of N is initialized as 1. The computer then multiply in line 40 the value of S (8) entered in line 20 with the present value of N (=1) in line 30 and stores it as P. The value of S, N and P are PRINTED in line 50. The LET statement in line 60 adds 1 to the present value of N as 1 and assigns new value to N as 2. In line 70 GOTO statement sends the control back to line 40. The LET statement in line 40 now multiply S with the new value of N and stores the new value of P as 16. In line 50 it PRINTS the value S, new value of N, and P. Next time N becomes 3 and P becomes 24, PRINTS recent values of S, N, and P and so on. The program will continue until you <Break> it to stop.

11.4 CONDITIONAL TRANSFER TO CONTROL

So far we have used GOTO statement unconditionally in the above programs. To make decision regarding program flow based on the result of an expression we transfer the control conditionally using IF ... THEN statement.

IF ... THEN statement

IF ... THEN statement is used to make decisions. IF … THEN statement with GOTO forms a **controlled loop** in which repetition is predefined. The repetition continues until certain condition is satisfied. It is illustrated in program 11.6.

PROGRAM 11.6

```
10    REM * GOTO LOOP OF WELCOME MESSAGE*
15    REM * PROGRAM STOPS WHEN A$ IS NAVEED *
20    INPUT "PLEASE ENTER YOUR NAME"; A$
30    PRINT
40    PRINT "WELCOME" ; A$
50    IF A$ = NAVEED  THEN GOTO 70
60    GOTO 20
70    PRINT "THERE IS A MESSAGE FOR YOU"
80    END
```

OUTPUT OF PROGRAM 11.6

```
PLEASE ENTER NAME? NISAR

WELCOME NISAR
PLEASE ENTER YOUR NAME? JAVEED

WELCOME JAVEED
PLEASE ENTER YOUR NAME? NAVEED

WELCOME NAVEED
THERE IS A MESSAGE FOR YOU
```

This program will continue to RUN as program 11.3, but as soon as the name NAVEED is entered in line 20, after executing line 30 and 40, the string variable is compared in line 50, when it finds A$ = NAVEED in line 50, it responds and GOTO statement brings the computer on line 70, which PRINTS the message after which the computer stops executing according to line 80 having END statement.

We can also stop executing the program for any desired value of N in program 11.6 using IF...THEN statement as illustrated in program 11.7 for the last value of N= 10.

PROGRAM 11.7

```
10    REM ** A TABLE OF NUMBERS FROM 1 TO 10**
20    INPUT "TABLE OF"   ; S
30    LET N = 1
40    LET P = S * N
```

```
50      PRINT S, N, P
60      LET N = N + 1
70      IF N >10 THEN GOTO 80
80      GOTO 30
90      END
```

OUTPUT OF PROGRAM 11.7

TABLE OF : ? 7 *(type any number for N say 7 after question mark ? and enter)*

```
7        1        7
7        2        14


7        10       70
```

As soon as N takes a value greater than 10, computer responds to it and GOTO statement in line 70 sends the control to line 90 to stop the execution of program. Thus the program 11.7 has a definite end unlike program 11.6 and breaks automatically as soon as N takes the desired value. (**Note** GOTO in line 70 is optional. It can be simply IF N >10 THEN 80 or IF N >10 THEN END.

PROGRAM 11.8

```
 10     REM * PRINTING A BIGGER Z OF CHARACTER WIDTH 25Zs *
 20     LET A = 0
 30     FOR I = 1 TO 3
 40     PRINT TAB(24) "ZZZZZZZZZZZZZZZZZZZZZZZZZ"
 50     NEXT I
 60     IF A = 1 THEN END
 70     LET N = 40
 80     FOR J = 1 TO 8
 90     PRINT TAB(N) "ZZZZZZZ"
100     N= N-2
110     NEXT J
120     A = 1
130     GOTO 30
```

OUTPUT OF PROGRAM 11.8

```
ZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZ
                  ZZZZZZZ
                ZZZZZZZ
              ZZZZZZZ
            ZZZZZZZ
          ZZZZZZZ
        ZZZZZZZ
      ZZZZZZZ
    ZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZ
```

11.5 IF..... THEN ..... ELSE

IF statement used in section 11.4 transfers control to some other part of the program provided the condition is true. IF ... THEN ... ELSE statement transfers control to another part of the program when the condition is false. The format of IF... THEN ... ELSE statement is:-

Line # IF (*condition*) THEN (*STATEMENT 1*) ELSE (*STATEMENT 2*)

When the condition is true, then statement 1 is executed THEN may be followed by a BASIC statement or a sequence of statements or it may be a line number for branching one or more statements. ELSE statement transfers control to *statement 2* if condition is false. In the absence of ELSE statement, control transfers automatically to next executable statement if the condition is false. Note that IF... THEN ... ELSE is a single statement. The program 11.9 illustrates the conversion of marks percentage into grades using IF... THEN ... ELSE statement.

PROGRAM 11.9

```
10    REM * GRADE FINDING FROM GIVEN PERCENTAGE MARKS *
20    PRINT "ENTER YOUR % MARKS:" ; M
30    IF M > = 90 THEN G $ = A : GOTO 70  ELSE
40    IF M > = 80     THEN G $ = B: GOTO 70
50    IF M > = 70 THEN G $ =C: GOTO 70  ELSE
60    IF M > = 60 THEN G $ =D  ELSE G $ =E
70    PRINT " YOUR GRADE IS:   " G $
80    END
```

OUTPUT OF PROGRAM 11.9

```
ENTER YOUR % MARKS : ? 82
YOUR GRADE IS     B
```

11.6 ON ... GOTO Statement (A Multiple-Choice Statement)

GOTO statement jumps unconditionally to the specified line number. If statement allows computer to follow one of the two possible branches depending upon whether the condition is true or false. Multiple branching is done by using ON ... GOTO statement that transfers control to any number of choices. ON ... GOTO has the format

Line #   ON   *Variable* GOTO *1Line #, 2Line #, 3Line #,* ......

Variable must be a positive number may be a numeric variable or arithmetical expression from 1 to 255. ON GOTO needs the variable as an integer. When the variable is 1 then control will jump to 1 line # when variable is 2 then to 2 line # and for 3 its jumps to 3 line # and so on.

PROGRAM 11.10

```
10   REM ** TO TELL THE NUMBER IS IN POSITIVE, NEGATIVE OR ZERO **
20   INPUT       " ENTER THE NUMBER PLEASE "; N
30   A$ = " NEGATIVE "
40   B$ = " ZERO "
50   C$ = " POSITIVE "
60   D$ = " THE NUMBER YOU ENTERED IS "
70   S = SGN (N) + 2
80   ON S GOTO 90, 100, 110
90   PRINT D$, A$
100  PRINT D$, B$
110  PRINT D$, C$
120  END
```

OUTPUT OF PROGRAM 11.10

```
ENTER THE NUMBER PLEASE ? 57
THE NUMBER YOU ENTERED IS POSITIVE
```

PROGRAM 11.11

```
10   REM
20   CLS
30   PRINT        " WHAT IS THE VALUE OF (2+4^2)/2+4 "
40   PRINT        " 1 ............ 22 "
50   PRINT        " 2 ............ 3 "
60   PRINT        " 3 ............ 13 "
70   PRINT        " 4 ............ 14 "
80   PRINT        " 5 ............ NONE OF THE ABOVE "
90   PRINT
100  INPUT       " WHAT IS YOUR ANSWER 1, 2, 3, 4 or 5 "; S
110  ON S GOTO 120, 160, 200, 230, 270
120  PRINT       " *********** WRONG ********** "
130  PRINT       " EXPONENTS ARE DONE FIRST "
140  PRINT       " ******** TRY AGAIN ******** "
150  GOTO 90
160  PRINT       " *********** WRONG ********** "
170  PRINT       " DIVISION IS DONE BEFORE ADDITION "
180  PRINT       " ******** TRY AGAIN ******** "
190  GOTO 90
200  PRINT       " WELL DONE! YOU KNOW HOW TO "
210  PRINT       " EVALUATE A BASIC EXPRESSION "
220  GOTO 310
230  PRINT       " *********** WRONG ********** "
240  PRINT       " PARENTHESES ARE EVALUATED FIRST "
250  PRINT       " ******** TRY AGAIN ******** "
260  GOTO 90
270  PRINT       " *********** WRONG ********** "
280  PRINT       " *********** SO POOR *********** "
290  PRINT       " ******** TRY AGAIN ******** "
300  GOTO 90
310  END
```

OUTPUT OF PROGRAM 11.11

```
WHAT IS THE VALUE OF (2+4^2)/2+4
1 ------- 22
2 ------- 3
3 ------- 13
4 ------- 14
5 ------- NONE OF THE ABOVE

WHAT IS YOUR ANSWER 1, 2, 3, 4 or 5 ? 1
********** WRONG **********
EXPONENTS ARE DONE FIRST
******** TRY AGAIN ********

WHAT IS YOUR ANSWER 1, 2, 3, 4 or 5 ? 2
********** WRONG **********
DIVISION IS DONE BEFORE ADDITION
******** TRY AGAIN ********

WHAT IS YOUR ANSWER 1, 2, 3, 4 or 5 ? 5
********** WRONG **********
********** SO POOR **********
******** TRY AGAIN ********

WHAT IS YOUR ANSWER 1, 2, 3, 4 or 5 ? 4
********** WRONG **********
PARENTHESES ARE EVALUATED FIRST
******** TRY AGAIN ********

WHAT IS YOUR ANSWER 1, 2, 3, 4 or 5 ? 3
WELL DONE! YOU KNOW HOW TO
EVALUATE A BASIC EXPRESSION
Ok_
```

11.7 ON ERROR GOTO

The ON ERROR GOTO statement suppresses error message, as soon as an error occurs, it transfers control to the line number specified. The ON ERROR OFF no longer suppresses error message i.e., the computer handles error in the normal way. The Report command prints out the error message for the last error.

PROGRAM 11.12

```
10   REM ** PROGRAM WITHOUT ERROR MESSAGES **
20    X= 0
30   PRINT 27/X
40   PRINT : PRINT "IMPOSSIBLE"
50   END
```

OUTPUT OF PROGRAM 11.12

```
Division by zero
 1.701412E+38

IMPOSSIBLE
```

Now using ON ERROR GOTO statement let us try the following program. The error number is stored in the variable ERR. The line number at which the error occurred is stored in the variable ERL.

PROGRAM 11.13

```
10   REM ** ERROR DETECTING PROGRAM **
20   ON ERROR GOTO 200
30   X = 0
40   PRINT 27/X
50   PRINT: PRINT "IMPOSSIBLE"
60   END
200  PRINT "ERROR CODE" ; ERR
210  PRINT "OCCURS AT LINE"; ERL
220  GOTO 50
```

OUTPUT OF PROGRAM 11.13

```
ERROR CODE 11
OCCURS AT LINE 40

Division by zero in 40
Ok
```

A disadvantage of the ON ERROR GOTO technique is that any error, including a disk problem or a syntax error, will cause the program to GOTO the line specified. This can make the debugging very difficult.

11.8 FOR ... NEXT loops

In some cases a part of a program is repeated several times. It is necessary to use a loop to execute the part of the program repeatedly. A combination of IF ... THEN and GOTO statements can be used to perform looping. Consider a standard finite loop of the form as follows. This loop executes 16 times.

```
   .
   .
   50 LET K = 1
   60
   .            Body of the Loop
   .
Loop
   200
   210 LET K = K + 1
   220 IF K < = 16 THEN GOTO 60
   230
   .            Rest of the Program
   .
```

Such loops are so common and useful in programming that BASIC has included special instructions such as FOR and NEXT to form counter loops in

such loops the execution of certain statement is repeated for a specified number of times. According to these instructions, the above loop may be written as:-

```
      .
      .
      50  FOR K = 1 TO 16        Name of the Counter, Initial and Final Value
  ┌─→ 60 ─────────────────────────────┐
  │   .                               │
L │   .                  Body of the Loop
o │   .                               │
o │   200 ────────────────────────────┘
p └── 220 NEXT K          Increment and Counter Test
      230 ────────────────────────────┐
      .                  Rest of the Program
      .
```

The purpose of FOR statement at the beginning of the loop is to name the counter and put its initial and final values. The NEXT statement indicates the end of the loop, increments the counter and then tests the counter. If the counter is less than or equal to 16 in line 220, then control goes to statement immediately after FOR statement (line 60 in the above case). If the counter is greater than 16 in line 220, then the **loop ends** and the program continues following the NEXT statement (line 230). The following program will illustrate FOR ... NEXT loops.

PROGRAM 11.14

```
10    REM ** USE OF COUNTER LOOP **
20    READ A$
30    FOR K = 1 TO 4
40    PRINT A$
50    NEXT K
60    DATA I LOVE PAKISTAN
70    END
```

OUTPUT OF PROGRAM 11.14

```
I LOVE PAKISTAN
I LOVE PAKISTAN
I LOVE PAKISTAN
.   .   .
.   .   .
```

As another example, suppose you want to PRINT odd-numbers between 1 to 100. This can be done as illustrated in program 11 using FOR... NEXT loop from the initial value 1 to the final value with an increment of 2.

PROGRAM 11.15

```
10    REM ** PRINTING OF ODD NUMBERS FROM 1 TO 100**
20    PRINT "  ODD NUMBER  "
30    FORK = 1 TO 100 STEP 2
40    PRINT K,
50    NEXT K
60    END
```

OUTPUT OF PROGRAM 11.15

```
ODD NUMBERS
1        3        5        7        9
11       13       15       17       19
.        .        .        .        .
.        .        .        .        .
91       93       95       97       99
```

The output appears in a line if semicolon is used instead of comma in 40.

PROGRAM 11.16

```
10   REM ** PRINTING OF ODD NUMBERS FROM 1 TO 100**
20   PRINT "  ODD  NUMBERS  "
30   FOR K = 1 TO 100 STEP 2
40   PRINT  K;
50   NEXT K
60   END
```

OUTPUT OF PROGRAM 11.16

```
ODD NUMBERS
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41
43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81
83 85 87 89 91 93 95 97 99
```

In Program 11.7 we have used controlled loop. It can be rewritten using a counter loop as illustrated in Program 11.17 using FOR ... NEXT statements

PROGRAM 11.17

```
10   REM ** A TABLE OF NUMBERS FROM 1 TO 10**
15   REM** USING COUNTER LOOP WITH COUNTER K**
20   FOR K = 1 TO 10
30   LET P = N * K
40   PRINT N, K, P
50   NEXT K
60   END
```

OUTPUT OF PROGRAM 11.17

```
TABLE OF: ? 9
9        1        9
9        2        18
.        .        .
.        .        .
9        10       90
```

The value of the increment specified by STEP may be negative, but then the initial value of the counter should be larger than the final value as illustrated in program 11.18, in which the value decreases by 3 in each step.

PROGRAM 11.18

```
10   REM**USING COUNTER FOR VALUES IN DESCENDING ORDER**
20   FOR N = 21 TO 0   STEP - 3
30   PRINT N;
40   NEXT N
50   END
```

OUTPUT OF PROGRAM 11.18

21 18 15 12 9 6 3 0

In program 11.18 NEXT N in line 30 assigns new value to the counter N as 3 less than the previous value. The loop is repeated as long as N > = 0 is true.

11.9 WHILE and WEND Statement

WHILE...... WEND statement is used to execute a series of statements in a loop as long as a given condition is true. If the condition is false, the loop ends and the program continues to the line following EEND (if there exists a line). WHILE.... WEND loop is similar to that of FOR ... NEXT loop. The differences between the two are that WHILE.... WEND loop is used when the number of iterations is not known in advance to the programmer and moreover it checks the looping condition and decides whether or not to continue as illustrated in Program 11.19, in which we calculate the factorial of a number N (ie. N! = N*(N-1)*(N-2) ... 2*1) and in program 11.20 where we find the members of a successive series using WHILE and WEND statements.

PROGRAM 11.19

```
10    REM ** PRINTING THE VALUE OF FACTORIAL " N " **
20    INPUT " ENTER THE NUMBER N "; N
30    F = 1 :  A = N
40    WHILE  N > 0
50    F = F * N
60    N = N - 1
70    WEND
80    PRINT  " FACTORIAL OF ";  A "  = "; F
90    END
```

OUTPUT OF PROGRAM 11.19

```
ENTER THE NUMBER N? 5
 FACTORIAL OF 5   = 120
```

PROGRAM 11.20

```
10    REM * PRINTING SUCCESSIVE VALUES OF A SERIES *
20    INPUT " ENTER THREE NUMBERS X, Y, R "; X, Y, R
30    WHILE  X < Y
40    PRINT  X,
50    X = X * R
60    WEND
70    PRINT  "*GOOD BYE*"
80    END
```

OUTPUT OF PROGRAM 11.20

```
ENTER THREE NUMBERS X, Y, R? 5,  500,  3
5              15             45             135            405
*GOOD BYE*
```

PRODUCING SOUND USING WHILE AND WEND LOOP

Sounds of different frequencies can be produced using BEEP and SOUND statements. Every beep sounds the same having the same interval of time.

PROGRAM 11.21

```
10    REM * BEEPING AFTER EVERY 2 SECONDS *
20    BEEP
30    START = TIMER
40    WHILE TIMER < START + 2
50    WEND
60    GOTO 10
70    END
```

To **produce sound** of frequency F and duration T, the syntax is

SOUND F, T

Note: **F** is the frequency in Hz., which must be a number from 37 to 32767. The variable **T** is the time interval described in ticks and may be a number from 0 to 65535 and need not be an integer its duration in ticks (18.2 ticks = 1 sec.) Run the following program and note the changes for different TIMER settings.

Program 11.22 produces two notes of frequencies 256 Hz. and 480 Hz. after every half second (9 ticks). Run this program with different frequencies.

PROGRAM 11.22

```
10    REM ** SOUND DEMONSTRATOR **
20    CLS
30    WHILE 1
40    SOUND 256, 9
50    SOUND 480, 9
60    WEND
```

SOUND EFFECTS USING FOR … NEXT LOOP

Try the following FOR … NEXT loop program that produces sound of increasing frequency from 200 Hz to 500 Hz. in a step of 1 Hz.(By default) with a duration of 0.1 tick

PROGRAM 11.23

```
10    REM ** SOUND OF 200Hz TO 500Hz WITH INCREASING PITCH**
20    FOR FREQ = 200 TO 500
30    SOUND FREQ,  .1
40    NEXT FREQ
50    GOTO 20
```

In the program 11.24, the FOR…NEXT loop produces a repetition of very short notes from 200 Hz. to 500 Hz. Thus you hear sounds of rapidly rising pitch. Now change line 20 so that the sound produced has a falling pitch from 500 to 200 Hz.. The frequency is decreasing by 1Hz after 0.1 tick as in program 11.22.

PROGRAM 11.24

```
10   REM ** SOUND OF 500Hz TO 200Hz WITH DECREASING PITCH**
20   FOR FREQ = 500 TO 200 STEP -1
30   SOUND FREQ, .1
40   NEXT FREQ
50   GOTO 20
```

Now put both ideas together into Program Siren Sound. As the name suggests, it makes a sound like a siren until you press CTRL+BREAK.

PROGRAM 11. 25

```
10   REM ** SIREN SOUND **
20   CLS
30   FOR R = 1 TO 8
40   REM * RISING PITCH INCREASING WITH STEP 1 AS DEFAULT *
50   FOR F = 200 TO 1000
60   SOUND F, .1
70   NEXT F
80   REM * FALLING PITCH DECREASING WITH STEP 2 *
90   FOR F  = 1000 TO 200 STEP -2
100  SOUND F, .1
110  NEXT F
120  NEXT R
130  END
```

You can make the pitch rise and fall at different rates by changing lines 50, 60, 90 and 100. Try various combinations of Step size and duration.

11.10 NESTED LOOPS: Loops within Loop:

FOR … NEXT loops may be nested; i.e. a FOR.... NEXT loop may be placed inside another FOR... NEXT loop. The following conditions must be met while nesting a FOR.... NEXT loop.

- Each loop must have a different variable name as its counter.
- The NEXT of inner loop must appear before the outer loop.
- If nested loops have the same end point, a single NEXT statement may be used for all of them.

PROGRAM 11.26

```
10   REM ** K LOOP IS NESTED WITHIN J LOOP **
20   FOR J = 1 TO 2
30   PRINT: PRINT " PAKISTAN "
40   FOR K = 1 TO 3
50   PRINT " ZINDA BAD "
60   NEXT K
70   NEXT J
80   END
```

OUTPUT OF PROGRAM 11.26

```
PAKISTAN
ZINDA BAD
ZINDA BAD
ZINDA BAD

PAKISTAN
ZINDA BAD
ZINDA BAD
ZINDA BAD
```

Suppose we want to write a program with output that repeats three times with a change of lot no each time. This repetition of the pattern can be translated into BASIC by writing a simple FOR.... NEXT nested loop to produce the desired pattern as follows:-

PROGRAM 11.27

```
10  REM ** J LOOP IS NESTED WITHIN I LOOP **
20  FOR I = 1 TO 3
30  PRINT: PRINT " LOT NO: " ; I
40  FOR J = 1 TO 4
50  K = 100 + 25 * J
60  PRINT K
70  NEXT J
80  NEXT I
90  END
```

OUTPUT OF PROGRAM 11.27

```
LOT NO: 1        125
                 150
                 175
                 200

LOT NO: 2        125
                 150
                 175
                 200

LOT NO: 3        125
                 150
                 175
                 200
```

WHILE ... WEND loops also be nested to any level each WEND will match the most recent "WHILE" statement. An unmatched "WEND" causes a "WEND without WHILE" error. For Labs. consult **Computer Lab. Journal** for SSC. Computer Science by Prof. M. Tahir Hassan.

SUMMARY

GOTO Statement

GOTO is one of the control statements that causes the control to transfer anywhere within the Program. The repetition of a loop may be infinite.

GOTO with READ ... DATA

Loops are particularly useful in programs containing READ.....DATA statements.

GOTO using LET statement

By using GOTO statement and loops, the computer performs tasks repeatedly, quickly and efficiently.

CONDITIONAL TRANSFER TO CONTROL

we transfer the control conditionally using IF ... THEN statement and make decisions regarding program flow based on the result of an expression.

IF ... THEN statement

IF ... THEN statement is used to make decisions. IF … THEN statement with GOTO forms a **controlled loop**. The repetition continues until certain condition is satisfied.

IF ... THEN ... ELSE

IF statement transfers control to some other part of a program provided certain condition is true. IF ... THEN ... ELSE statement transfers control to another part of the program if the condition is false. In the absence of ELSE statement, control transfers automatically to next executable statement if the condition is false.

ON … GOTO Statement (A Multiple-Choice Statement)

Multiple branching is done by using ON … GOTO statement that transfers control to any number of choices.

ON ERROR GOTO

The ON ERROR GOTO statement suppresses error message. As soon as an error occurs, it transfers control to the line number specified. The ON ERROR OFF no longer suppresses error message and the computer handles error in the normal way. The Report command prints out the error message for the last error.

FOR ... NEXT loops

In some cases a part of a program is repeated several times. It is necessary to use a loop to execute the part of the program repeatedly for a given number of times. FOR and NEXT forms a counter loop in which the execution of certain statement is repeated for a specified number of times.

WHILE and WEND Statement

WHILE...... WEND statement is used to execute a series of statements in a loop as long as a given condition is true. WHILE.... WEND loop is similar to that of FOR.... NEXT loop. The differences between the two are that WHILE.... WEND loop is used when the number of iterations is not known in advance to the programmer and moreover it checks the looping condition and decides whether or not to continue.

NESTED LOOPS: Loops within Loop:

FOR … NEXT loops may be nested; i.e. a FOR ... NEXT loop may be placed inside another FOR ... NEXT loop. The following conditions must be met while nesting a FOR ... NEXT loop.

- Each loop must have a different variable name as its counter.
- The NEXT of inner loop must appear before the outer loop.
- If nested loops have the same end point, a single NEXT statement may be used for all of them.

WHILE ... WEND loops also be nested to any level each WEND will match the most recent "WHILE" statement.

EXERCISE

11.01 Complete the following statements

i) The order in which program statements execute can be changed using a ______________ statement.

ii) The last line of the output using GOTO loop with READ...DATA statement is ________________________.

iii) In looping a statement is executed _______________.

iv) IF ... THEN statement with GOTO forms a _____________ loop.

v) ________________ statements are used to form counter loop.

11.02 Tick ( ) the following statements either True or False.

i) The decision block controls the statement to be executed True/False

ii) GOTO is a conditional statement. True/False

iii) STOP statement is used to terminate an endless loop. True/False

iv) GOTO loop perform repeated tasks quickly and efficiently. True/False

v) The statement " FOR K = 1 TO 10 STEP -2 " is incorrect. True/False

11.03 Encircle only one choice A, B, C or D in each case.

i) Which of the following statements can be used for unlimited looping?
(A) GOTO (B) FOR ... NEXT
(C) IF THEN (D) ON GOTO

ii) Which of the following statements transfers control unconditionally?
(A) GOTO (B) FOR ... NEXT
(C) IF THEN (D) ON GOTO

iii) Which of the following statements can be used for counter loop?
(A) GOTO (B) FOR ... NEXT
(C) IF THEN (D) WHILE ... WEND

iv) Which of the following statements can be used for multiple branching?
(A) GOTO (B) IF THEN ELSE
(C) ON GOTO (D) WHILE ... WEND

v) ON ERROR GOTO statement is used to detect
   (A) Logical Errors (B) Syntax Errors
   (C) Execution Errors (D) Logical and Syntax Errors

vi) Name the error that you think to exist in the formula C = C*D/C
   (A) Logical (B) Syntax (C) Execution (D) None

11.04 Match the items given in Column I with those in Column II

Column I	Column II
i) GOTO loop	a) Multiple choice
ii) IF … THEN	b) WHILE N > 5
iii) FOR … NEXT	c) N = N + 2
iv) Nested loop	d) ENDLESS
v) WEND	e) N = 1 TO 5
vi) ON … GOTO	f) N >= 5
	g) Loops within loop

11.05 Predict the output, or give the ERROR message for each of the following programs:

a)
```
10 READ A, B
20 PRINT A&B
30 GOTO 10
40 DATA. 3,8,6,4
50 END
```

b)
```
10 READ A, B
20 PRINT A &B
30 DATA 5, 7, 8, 4
40 GOTO 20
50 END
```

c)
```
10 LET X = 17
20 IF (X< 3) OR (X >18) THEN PRINT "YES"
30 PRINT "NO"
40 END
```

d)
```
10 LET X = 23
20 IF NOT (X > 35) THEN GOTO 50
30 PRINT "KNOT"
60 END
```

e)
```
10  LET J = - 3
20  PRINT J + 2
30  PAINT "ABC"
40  LET J = J + 1
50  IF J < = 1 THEN GOTO 20
60  END
```

11.06 What is transfer of control? Give an example which causes transfer of control.

11.07 What is the difference between conditional and unconditional transfer of control? Give example of each type.

11.08 How do FOR…NEXT statement differs from IF…THEN statement?

11.09 How do IF…THEN statement differs from IF…THEN…ELSE statement?

11.10 What is multiple branching? Write a program to explain multiple branching using ON GOTO statement.

11.11 What is the difference between controlled loops and counter loops?

11.12 What is the purpose of FOR statement in the FOR…NEXT loop. Explain with example.

11.13 What is the function of the NEXT statement in the FOR…NEXT loop? Explain with example.

11.14 How do WHILE…WEND loop differ from FOR…NEXT loop?

11.15 What is the function of WEND in WHILE…WEND statement?

11.16 Explain the use of ON…ERROR GOTO statement.

11.17 What is meant by nested loop(s)? Illustrate the use of Nested loop using FOR…NEXT statement.

12

ARRAYS

Many programming applications use lists or tables. Mailing lists, lists of students, their result sheets, employees salary sheets are all items for which a computer program might require the use of large numbers of different variables. Given below are the partial list and a result sheet of students in a class.

A list of students

FAHEEM
KAMRAN
ABID
AHMAD
HARIS

Table 12.1 Result of students in a test

Roll #	English	Urdu	Maths
14	72	79	82
15	68	72	76
16	54	59	95
17	81	73	90
18	63	59	84

As each item is assigned a different variable name, hence it would be a tedious task to type so-many large number of different variables and also number of LET, INPUT, or READ statements. BASIC provides an easy way to handle such situations in which numbers and strings are assigned to many different variables called subscripted variables. Lists and tables are called **arrays**. A list is referred to as one-dimensional array and a table as two-dimensional array.

12.1 ARRAYS

Items in a list or in a table are represented by a single variable called name of the array. **An array is a set of like variables arranged in sequence identified by their subscripts** or **an array is a collection of subscripted variables with the same variable name**. The subscripted variable names are similar to simple variables names except that they are followed by a subscript. **The subscript must be either zero or a positive integer and must always be enclosed in parentheses**. For example A (2), A (5), B$ (4), NUM (0), NAME$ (7).

In the above examples A is a variable name while 2 or 5 are the subscripts. Similarly B$ is a string variable name and 4 is its subscript and so on. Usually the biggest number that a computer allows automatically as a subscript is 10. A

computer must be instructed to allow subscripts more than 10. Consider an array named " **A** " with members A(0), A(1), … A(6) as illustrated below.

Each subscripted variable is used to store one number or string. Values are assigned to the subscripted variables by LET, READ or INPUT statements. The use of a subscripted variable say A(2) in a program automatically reserves 11 memory locations (*for subscripts 0 to 10*) for **Array A** for elements A(0) to A(10). However we do not need to use element with subscript 0 if the number of elements is less than 11 though it is available.

Array A

A (0)	63
A (1)	42
A (2)	111
A (3)	75
A (4)	123
A (5)	98
A (6)	55

BASIC also allows the use of text or string variable arrays having strings values similar to simple string variables. Consider an array ART$ having string values. Here the variable ART$(2) is assigned the string "BOOK". The instruction PRINT ART$(2) will produce the string BOOK as output.

ARRAY ART$

ART$ (1)	PEN
ART$(2)	BOOK
ART$ (3)	PENCIL
ART$ (4)	RUBBER
ART$ (5)	FILE

12.2 FILLING AN ARRAY

Strings or numbers are assigned to subscripted variables in arrays using LET READ or INPUT statements. It is called filling an array. One way to fill this array is illustrated by PROGRAM 12.1 using READ ... DATA statements.

PROGRAM 12.1

```
10    REM ** ARTICLES **
20    READ ART $(1), ART$(2), ART$ (3), ART$ (4), ART$ (5)
30    DATA PEN, BOOK, PENCIL
40    DATA RUBBER, STAMP
50    END
```

There does not seem to be a need for subscripted variables. However, using the idea that the subscript of a subscripted variable may itself be a variable, let us write PROGRAM 12.2 using a FOR ..... NEXT loop.

PROGRAM 12.2

```
10       REM ** ARTICLES, VERSION 2 **
20       FOR I = 1 TO 5
30       READ ART $ (I)
40       NEXT I
50       DATA PEN, BOOK, PENCIL
60       DATA RUBBER, STAMP
70       END
```

When this PROGRAM is executed, the subscripted variable ART$(I) successively represents all the elements in the array. The first time through the

loop when I=1, the string PEN is assigned to ART$(I). The second time through the loop; Book is assigned to ART$(2) and so on.

PROGRAM 12.2 does exactly the same thing as PROGRAM 12.1, but is two lines longer. Suppose the DATA list contained 50 strings rather than 5 then the method used in the first PROGRAM would need to type 50 different variables ART$(1) ART$(2).... ART$(50). While the technique of using a FOR... NEXT loop and subscripted variables to fill the array would save a lot of typing.

12.3 PRINTING AN ARRAY

Once string or numbers are stored in an array, the PROGRAM can manipulate them in many ways. Here is a PROGRAM that fills an array with a list of marks obtained by a student in various tests.

PROGRAM 12.3

```
10    REM ** ** MARKS SHEET IN ASCENDING ORDER **
20    FOR J = 1 TO 4
30    READ TEST (J)
40    NEXT J
50    FOR N = 1 TO 4
60    PRINT "  MARKS IN TEST # "; N; "     =    "; TEST (N)
70    NEXT N
80    DATA 384, 414.397,405
90    END
```

OUTPUT OF PROGRAM 12.3

```
MARKS IN TEST # 1 = 384
MARKS IN TEST # 2 = 414
MARKS IN TEST # 3 = 397
MARKS IN TEST # 4 = 405
```

In the first FOR ... NEXT loop of this PROGRAM, the numbers on the DATA list are assigned in order to the subscripted variables TEST(1), TEST(2), TEST(3) AND TEST(4). Then since the counter M also goes from 1 to 4 in the second FOR ... NEXT loop, the values of the subscripted variables are printed in line 60. Note that the same counter can be used both FOR... NEXT loops, since the loops do not overlap.

PROGRAM 3 is a simple PROGRAM to illustrate filling and PRINTing an array. Some processing may also be required with the subscripted variables in an array before PRINTing. For example to PRINT the same report in reverse order. PROGRAM 3 is rewritten in revised version.

PROGRAM 12.4

```
10    REM ** MARKS SHEET IN DESCENDING ORDER **
20    FOR J = 1 TO 4
```

```
30   READ TEST (J)
40   NEXT J
50   FOR N = 4 TO 1 STEP -1
60   PRINT "MARKS IN TEST #"; N; " = "; TEST(N)
70   NEXT N
80   DATA 384, 414.397,405
90   END
```

OUTPUT OF PROGRAM 12.4

```
MARKS IN TEST # 4 = 405
MARKS IN TEST # 3 = 397
MARKS IN TEST # 2 = 414
MARKS IN TEST # 1 = 384
```

12.4 BIG ARRAYS: DIM

DIM is a short name used for **DIM**ension. DIM statement is used to **declare an array** especially to specify more than 10 subscripts. DIM statement causes the computer to reserve more spaces in its memory required for big arrays. Let us modify PROGRAM 12.2 having 20 elements as shown in program 12.5

PROGRAM 12.5

```
10   REM ** AN ARRAY OF 20 ARTICLES **
20   FOR I = 1 TO 20
30   READ ART$ (I)
40   NEXT I
50   DATA PEN, BOOK ... (total of 20 DATA values)
60   END
```

OUTPUT OF PROGRAM 12.5

```
BAD SUBSCRIPT ERROR IN 30
```

The reason that the BAD SUBSCRIPT ERROR (or SUBSCRIPT OUT OF BOUNDS) occurs in line 30 is that the computer ordinarily reserves memory spaces for only 11 elements of an array. Whereas in PROGRAM 12.5 twenty subscripted variables are needed. The number of subscripted variables can be increased including the DIM statement in an array program. For example the format of the DIM statement to increase the array elements ART$ in the above PROGRAM 12.5 will be:-

Line # **DIM ART$ (20)**

The following program 12.6 can tell you the maximum number for a subscripted variable that is acceptable by your computer.

PROGRAM 12.6

```
10   REM ** DIMensioning
20   INPUT N1
```

```
30    DIM BIG (N1)
40    PRINT "OKAY"
50    END
```

Run the program and type a large number. An OKAY message means the computer has accepted the number as a DIMension otherwise it will print an error message of some kind. By experimenting find the biggest number that will be accepted as DIMension.

12.5 TWO DIMENSIONAL ARRAYS

The arrays we have discussed so far are all one dimensional arrays. Sometimes data is arranges in rows and columns. Thus two or more subscripts are needed to represent each element in such arrays. For example the table 12.2 below summarizes the attendance of Sir Syed School on 23rd December 2002.

Table 12.2 Attendance of Sir Syed School, on 23rd December 2002

CLASS	Number of Students		
	PRESENT	LEAVE	ABSENT
VI	217	13	28
VII	198	14	24
VIII	203	16	21
IX	226	18	13
X	214	11	9

The location of the data in this table can be described most conveniently by numbering both rows and columns as given below. For example the location of the students on leave in class VIII is described by " row 3 and column 2 ".

Columns

Rows	1	2	3
1	217	13	28
2	198	14	24
3	203	16	21
4	226	18	13
5	214	11	9

	COL. 1	COL. 2	COL. 3
ROW 1	S (1, 1) 217	S (1, 2) 13	S (1, 3) 28
ROW 2	S (2, 1) 198	S (2, 2) 14	S (2, 3) 24
ROW 3	S (3, 1) 203	S (3, 2) 16	S (3, 3) 21
ROW 4	S (4, 1) 226	S (4, 2) 18	S (4, 3) 13
ROW 5	S (5, 1) 214	S (5, 2) 11	S (5, 3) 9

The location of the data in this table can be described most conveniently by numbering both rows and columns. Two-dimensional arrays are named the same way as one-dimensional arrays. The elements in a two-dimensional array are commonly referred to as doubly-subscripted variables.

DIMensioning a Two-Dimensional Array

When a doubly-subscripted variable is used in a program, then the number of subscripted variables can be increased as for single-subscripts using DIM statement. The format of the DIM statement will be:-

Line # DIM (*Array Name*) (*row, Column*)

For example, if the data to be assigned to the variables in an array P is arranged in 30 rows and 15 columns, then the DIM statement will be:-

Line # DIM P(30, 15)

Actually the above DIM statement for array P reserves memory space for data in 31 rows, numbered 0 to 30, and 16 columns, numbered 0 to 15. when using two-dimensional arrays however, the number 0 is often avoided as a subscript. When more than one array will be used in a program, the arrays may all be DIMensioned in the same statement; for example,

DIM ST$ (20, 35), NUM (46, 12)

12.6 FILLING AND PRINTING TWO-DIMENSIONAL ARRAYS

One way of filling a two-dimensional array is by FOR … NEXT nested loops. Let's write a program to fill the array D with the numbers representing the results of the dress code poll. One possible outline of an algorithm is:-

Step 1. Dimension the array D to 4 rows and 3 columns.
Step 2. Arrange the numbers from the table row by row in DATA statements
Step 3. For each row, READ all the columns.

There are many ways the numbers from the table can be arranged in DATA statements, but usually they are typed row by row such as given bellow.

```
DATA 158, 100, 30
DATA 131, 95, 26
DATA 125, 130, 20
DATA 110, 124, 22
```

Then in Step 3, to READ all the numbers (columns) across each row, we can have the computer READ the variable D(ROW, COL) in the following order:

```
FOR ROW = 1
    COL = 1        COL = 2 COL = 3
FOR ROW = 2
    COL = 1        COL = 2 COL = 3
      :              :       :
```

This is exactly the order specified by these nested FOR … NEXT loops:-

```
FOR ROW = 1 TO 4
                        FOR COL = 1 TO 3
```

```
:
NEXT COL
NEXT ROW
```

Study the following program of filling two dimensional array:

PROGRAM 12.8

```
10    PRINT "FILLING A TWO-DIMENSIONAL ARRAY"
20    DIM D(4, 3)
30    FOR ROW = 1 TO 4
40    FOR COL = 1 TO 3
50    READ D(ROW, COL)
60    NEXT COL
70    NEXT ROW
200   DATA 158, 100, 30
210   DATA 131, 95, 26
220   DATA 125, 130, 20
230   DATA 110, 124, 22
240   END
```

The data in line # 200 – 230 may be entered in a single line # 200 as:-

```
200   DATA 158, 100, 30, 131, 95, 26, 125, 130, 20, 110, 124, 22
```

Although a single line is more difficult to correct and retype data than in separate statements. The important thing is the order in which the data appear.

After filling array D, let's PRINT the array in **table form** with row and column numbers. Following statements are to be added in the above program:-

```
80    PRINT "HERE'S THE TABLE:"
90    PRINT "ROW / COL"; TAB (10); "
```

12.7 MANIPULATING DATA WITH ARRAYS

Numerous operations on data can be performed using arrays, including finding a particular element in an array, matching elements from two arrays and sorting an array. To sort an array, is to rearrange the list from smallest to largest or vice versa. To demonstrate another such operation, consider the problem of finding the largest value in an array. To find the largest value in a numeric array, we use a FOR loop to search through the array.

An important use of arrays is to store data that are used later for comparison. These comparisons frequently involve searching an array to find the element that matches a particular value. Often referred to as "**table look-up**". This procedure may involve finding a life insurance premium for a given person's age, finding the grade of a student etc.

SUMMARY

Many computer programs might require the use of large numbers of variables. BASIC provides an easy way to handle such situations in which the data is in the form of list or table called **arrays**. Values of the data elements are assigned to variables called subscripted variables. A list is referred to as one-dimensional array and a table as two-dimensional array. Each subscripted variable is used to store one number or string.

FILLING AN ARRAY

Strings or numbers in a data are assigned to subscripted variables in arrays using LET READ or INPUT statements. It is called filling an array.

PRINTING AN ARRAY

Once string or numbers are stored in an array, it can be manipulated in many ways. One way is to use PRINT statement in a FOR ... NEXT loop .

BIG ARRAYS: DIM

DIM is a short name used for DIMension. DIM statement is used to specify subscripts greater than 10. It causes the computer to reserve more spaces in its memory for big arrays.

TWO DIMENSIONAL ARRAYS

Sometimes data is arranges in rows and columns. Thus two or more subscripts are needed to represent each element in such arrays. The location of the data in a table can be described easily by numbering both rows and columns. The variables of a two-dimensional array are as doubly-subscripted variables.

FILLING AND PRINTING TWO-DIMENSIONAL ARRAYS

One method of filling a two-dimensional array is nested FOR...NEXT loops.

MANIPULATING DATA WITH ARRAYS

Data can be manipulated with arrays performing operations on data. These include; finding a particular element in an array, matching elements from two arrays and sorting an array. To sort an array, is to rearrange the list from smallest to largest or vice versa.

An important use of arrays is to store data that are used later for comparison. These comparisons frequently involve searching an array to find the element that matches a particular value.

EXERCISE

12.01 Complete the following statements

i) A list of like variables identified by their subscripts is called as ______ array.

ii) A program to assign values to subscripted variables is called as _____ of array.

iii) The maximum number of variables accepted automatically in BASIC is ______ .

iv) The statement DIM A (50) would reserves ______ location.

v) DIM is a short name used for ____________________.

12.02 Tick (✓) the following statements either True or False.

i) An array is a collection of subscripted variables with the same variable name — True/False

ii) NUM 7 is a subscripted variable. — True/False

iii) A two dimensional array needs two subscripts to identify each element. — True/False

iv) BASIC automatically assigns 10 elements to a list of array. — True/False

v) DIM statement is used only in two dimensional arrays — True/False

12.03 Encircle one choice A, B, C or D in each case.

i) The number of elements assigned automatically to a two dimensional arrays in BASIC are:-

(A) 10 (B) 100 (C) 110 (D) 121

ii) The number of arrays that can be dimensioned in a DIM statement may be:-

(A) 1 (B) 2 (C) 4 (D) Any number

iii) The statement to reserve memory locations for an array M arranged in 12 columns and 20 rows may be:

(A) DIM (12, 20) (B) DIM M (20, 12)
(C) M (240) (D) DIM M (12, 20)

iv) Study the following program and predict the output on execution:-

```
10 FOR K = 1 TO 20
20 READ M (K)
30 NEXT K
40 DATA 20 1,8, 12, 13, 6, 19, 23
50 DATA 17, 5, 28, 14, 25, 11
60 END
```

(A) DATA OUT OF BOUND IN 40 (B) DATA OUT OF BOUND IN 50
(C) BAD SUBSCRIPT ERROR IN 20 (D) Ok

v) The subscript of a variable may be a
(I) Number (II) Alphabetic Character (III) Special Character
(A) I or II only (B) I only (C) I or III only
(D) All I, II or III

12.04 Match the items given in Column I with those given in Column II

	Column I		Column II
i)	LIST	a)	DIM
ii)	TABLE	b)	DIM P (20, 15)
iii)	FOR N = 1 TO 15	c)	DIM N (15, 20)
iv)	BIG ARRAY	d)	TWO DIMENSIONAL
v)	FOR COL = 1 TO 20	e)	ONE DIMENSIONAL
vi)	FOR ROW = 1 TO 20	f)	DIM A$ (20)

12.05 Predict the output of each of the following PROGRAM.

a)
```
10 FOR F = 1 TO 7
20 LET NUM (F) = F-1
30 NEXT F
40 FOR G = 1 TO 7
50 PRINT NUM (G)
60 NEXT G
70 END
```

b)
```
10 DIM N (11)
20 FOR P = 1 TO 11
30 LET N (P) = P
40 NEXT P
50 FOR T = 1 TO 11 STEP 2
60 PRINT N (T)
70 NEXT T
80 END
```

12.06 What is an array? Illustrate the difference with examples between one dimensional and two dimensional arrays.

12.07 What is the difference between simple and subscripted variables? Explain with examples.

12.08 What is the use of DIM statement? When it is needed?

12.09 What is the difference of DIM statement in one dimensional and two dimensional arrays?

12.10 Write a PROGRAM that fills an array with 16 random integers each between 25 and 60, then prints the array.

12.11 Write a PROGRAM that fills an array P with 20 numbers READ from a DATA list, then PRINTS the product of the elements of the array.

13

SUB-PROGRAM AND FILE HANDLING

When programs become sufficiently complex and require loops and decisions, the programmer feels the need of structured programming. A structured program is written in modules or blocks and allows only one entrance to each module and only one exit from each module. A program designed in this way is easy to follow and understand. In addition structured programs are much easier to correct than those having multiple entry and exit point to and from a module.

13.1 SUB-PROGRAMS

The primary idea of modular programming is to break up large problem into smaller, manageable parts called **modules** or **subprograms**. A subprogram is a group of statements that performs a single operation on execution. They are easy to design, debug, modify and document. Subprograms are written once in a main program but may be called at several places in the program. There are two types of subprograms:-

i) Function subprograms and
ii) Subroutine subprograms.

13.2 FUNCTIONS

Most Programming languages including BASIC contain functions that perform arithmetic and string operations. These functions make the programming much simpler and also reduce the labour and save time in programming.

INPUT VALUE	→	FUNCTION ROUTINE	=	OUTPUT VALUE

A function is a routine that takes a value (string or number) as an input, and produces a single value as output. There are two types of functions:-

- System-defined functions.
- User-defined functions.

13.3 SYSTEM DEFINED FUNCTIONS/BUILT-IN-FUNCTIONS

BASIC includes a wide range of **built-in-functions** in addition to the operations/functions already described in chapter 10. These built-in-functions are also called **library functions** or **system's defined functions** and are used in computer programs. Library functions are divided into two general categories; Numeric Functions and String Functions.

13.4 NUMERIC FUNCTIONS

These include ABS, INT, FIX, RND, SQR, LOG, EXP, SIN, COS, TAN etc. We shall explain only some very important BASIC Library functions. These functions can be executed in the direct mode as well as in the programming mode.

ABS FUNCTION

The purpose of ABS function is to provide the absolute value of the expression X i.e. the value without any sign. The format is; " ABS (X) " as illustrated by the following program 13.1. When you run the program, it asks the value for the variable. Enter different values after the prompt and note the output.

PROGRAM 13.1

```
10    REM ** GETTING ABSOLUTE VALUE OF A NUMBER **
20    INPUT " TYPE A NUMBER : " ; X
30    PRINT " ABSOLUTE VALUE IS :  " ABS (X)
40    END
```

OUTPUT OF PROGRAM 13.1

```
TYPE A NUMBER :? – 4.14
ABSOLUTE VALUE IS :  4.14
```

CINT FUNCTION

The CINT function converts the numeric expression to an integer by rounding the fractional part of the expression. CINT differs from the FIX & INT function, which returns integers by truncating the fractional part. Its format is:-

CINT (X)

INT FUNCTION

The INT function is called as integer function. This function prints the greatest integer less than or equal to the value in the parentheses i.e. positive whole numbers remain the same regardless of their decimal value and negative whole numbers are rounded down to the next smaller value. Its general format is:-

INT (X)

Where X is any number

RND FUNCTION

The RND function is referred to as a "Random number generator". It generates a number between 0 and 1. Its format is; " RND " or " RND (X) ". The value (X) inside the parentheses is optional. The value of X is used as a "**seed value**" by the computer to produce a random number. It has no effect on the range of the values produced by the random-number function. If the value 0 is the argument of RND function, then the last number generated by the function will be repeated. Execution of RND function is illustrated in program13.2.

PROGRAM 13.2

```
10    FOR X = 0 TO 3
20    FOR K = 1 TO 6
30    PRINT RND (X),
40    NEXT K
50    NEXT X
```

OUTPUT OF PROGRAM 13.2

```
.3116351    .3116351        .3116351    .3116351    .3116351
.3116351    .1213501        .651861     .8688611    .7297625
.798853     7.369805E-02                .4903128    .4545189
.1072496    .9505102        .7038703    .5318641    .9711614
.3209329    .9561278        .9345151    .5349368    .5644215
Ok_
```

SGN FUNCTION

The SGN function is referred to as the sign function, which reports the SIGN of a number and which can be equal to -1, 0 and 1. The general form is

SGN (X)

If the value of X is negative, the value returned by SGN (X) will be -1. If X is positive, SGN (X) will be 1.If X is equal to 0, SGN (X) returns to 0.

FIX FUNCTION

The purpose of this function is to obtain an integer value by simply dropping off the decimal point and numbers to the right of it. FIX does not round a number; it just ignores anything to the right of the decimal point. The format is:-

FIX (X)

When you run the program 13.3, it asks the value for the variable. Enter different values after the prompt and note the integer value as the output.

PROGRAM 13.3

```
10    REM ** GETTING  INTEGER  VALUE OF A NUMBER **
20    INPUT " TYPE A NUMBER : " ; X
30    PRINT " INTEGER VALUE IS :   " FIX (X)
40    END
```

OUTPUT OF PROGRAM 13.3

```
TYPE A NUMBER :? – 6.83
INTEGER VALUE IS : – 6
```

SQR FUNCTION

The purpose of this function is to find the square root of any positive number. Its format is; " SQR (X) " as illustrated in program 13.4. On execution, it prints the numbers form 1 to 5 successively in the first zone and their square roots in the second zone. You can find the square root of a single number.

PROGRAM 13.4

```
10    REM ** GETTING SQUARE ROOT OF NUMBERS FROM 1 TO 5 **
20    FOR X = 1 TO 5
30    PRINT X, SQR (X)
40    NEXT X
50    END
```

OUTPUT OF PROGRAM 13.4

```
1    1
2    1.414214
3    1.732051
4    2
5    2.236068
```

LOG FUNCTION

In BASIC, LOG function is used to find the value of natural logarithm (LOG in BASIC is a logarithm to the base e, where e = 2.718282). The format is:-

LOG(X)

Let us run program 13.5, it asks the value for the variable X. Enter different values after the prompt and note the output.

PROGRAM 13.5

```
10    REM ** GETTING  NATURAL LOG OF A NUMBER **
20    INPUT " TYPE A NUMBER : " ; X
30    PRINT " NATURAL LOG OF "; X ; "  IS   " ; LOG (X)
40    END
```

OUTPUT OF PROGRAM 13.5

```
TYPE A NUMBER : ? 10
NATURAL LOG OF  10  = 2.302585
```

SIN FUNCTION

The purpose of SIN function is to find the trigonometric ratio called sine of an angle X expressed in radians. Its format is:-

SIN (X) (*When X is in radians*)
SIN (3.1416* X/180) (*When X is in degrees*)

Type the following instruction for execution in the direct mode:-

? SIN (3.1416 * 45/180)

On pressing the <**ENTER**> ↵ key, the output will be:-

```
0.707
```

When you run SIN function in program 13.6, it asks the value for the variable. Enter different values after the prompt and note the output.

PROGRAM 13.6

```
10    REM ** GETTING THE SIN OF  AN ANGLE (IN RADIANS) **
20    INPUT " TYPE THE ANGLE IN RADIANS: " ; X
30    PRINT " SIN " ; X; "radians  = " ;  SIN (X)
40    END
```

OUTPUT OF PROGRAM 13.6

```
TYPE THE ANGLE IN RADIANS: ? 2
```

COS FUNCTION

The purpose of COS function is to find the trigonometric ratio called cosine of an angle X expressed in radians. The format is:

COS (X) (*When X is in radians*)

COS (π * X/180) (*When X is in degrees*)

For examples:-

PRINT COS (1)

As you press <**ENTER**>↵ key, you will get the output as:-

```
0.540302
```

PRINT COS (π * 30/180)

As you press <**ENTER**>↵ key, you will get the output as:-

```
0.866025
```

TAN FUNCTION

The purpose of TAN function is to find the trigonometric ratio called tangent of an angle X expressed in radians. The format is:-

TAN (X) (*When X is in radians*)

TAN (π * X/180) (*When X is in degrees*)

For examples:-

PRINT TAN (4)

As you press <**ENTER**>↵ key, you will get the output as:-

```
1.157821
```

PRINT TAN (π * 45/180)

As you press **<ENTER>**↵ key, you will get the output as:-

```
1
```

13.5 STRING FUNCTIONS

These include HEX$, LEFT$, MID$, CHR$, STR$, TIME$, INKEY$ etc.

HEX$ FUNCTION

HEX$ function calculate Hexadecimal number of given decimal number X (Where X can be a number from 0 to 65535). Its format is; HEX$ (X). For example:-

```
20 PRINT HEX$ (20)
```

RUN the above program that would print 14 (which is in Hexadecimal).

LEFT$ FUNCTION

LEFT$ function is used to select n left most characters of a given string. Its format is; " LEFT $ (A$, n) " as illustrated in program 13.7.

PROGRAM 13.7

```
10  REM ** PRINTING OF FIRST THREE CHARACTERS OF A STRING**
10  CLS
20  LET A$ = "PAKISTAN"
30  PRINT LEFT$ (A$, 3)
40  END
```

OUTPUT OF PROGRAM 13.7

```
PAK
```

RIGHT$ FUNCTION

RIGHT$ function selects rightmost " n " characters of a given string. Its format is; " RIGHT$ (X$, n) "

INKEY$ FUNCTION

INKEY$ function responds to any key from the keyboard without pressing the ENTER key. The statement is of the form described below in line no. 40 and can be included in any program.

```
:
30  PRINT "PRESS ANY KEY TO CONTINUE"
40  LET A$ = INKEY $
50  IF A$ = "  " THEN GOTO 40
:
```

SPACE FUNCTION

The purpose of SPACE$ function is to leave or skip certain number of spaces before a character . Spaces may be in the range 0 to 255. Its format is:-

SPACE$ (n) or SPC (n) (*Where* n *is the number of spaces*)

It is illustrated in programs 13.8 and 13.9

PROGRAM 13.8

```
10 PRINT " NAME: " ; SPACE $ (21); "ROLL NO:"
20 PRINT " CLASS: " ; SPC(20); "SECTION:"
```

OUTPUT OF PROGRAM 13.8

```
NAME:                    ROLL NO:
CLASS:                   SECTION:
```

PROGRAM 13.9

```
10    CLS
20    FOR N = 1 TO 5
30    LET A$ = SPACE$(2*N)
40    PRINT A$; N
50    NEXT N
60    END
```

OUTPUT OF PROGRAM 13.9

```
 1
   2
     3
       4
         5
```

13.6 USER DEFINED FUNCTIONS

Besides providing the built-in-functions, BASIC has the statement DEF FN that lets you to define your own functions. These are called user-defined functions. One of the advantage of a user-defined function is that once the function is defined, it may be used anywhere in the program in the same way in which built-in-functions are used. Another advantage is that a user-defined function can easily be changed to another function. Consider a relation between X and Y given below.

$$Y = X^2 + 3X$$

To find the value of Y for different values of X, we shall define a function FNS (X) as follows:-

DEF FNS (X) = X * X + 3 * X

Here 'S' is the function's name, X is the variable and X*X + 3 * X is the BASIC numerical expression that defines the function. Thus for X as 6, the function FNS (X) will give a numerical value 54 as follows:-

FNS (6) = 6 * 6 + 3 * 6 = 54

The DEF FN statement must appear before it is used in a program to define a function. It is illustrated in program 13.10.

PROGRAM 13.10

```
10   PRINT "USER - DEFINED FUNCTION"
20   DEF FNS (X) = X * X + 3 * X
30   PRINT " X ", "FNS (X)"
40   FOR K = - 3 TO 3
50   PRINT K, FNS (K)
60   NEXT K
70   END
```

OUTPUT OF PROGRAM 13.10

```
USER-DEFINED FUNCTION
      X              FNS (X)
     -3              0
     -2              -2
     -1              -2
      0              0
      1              4
      2              10
      3              18
```

13.7 SUBROUTINE SUBPROGRAMS

A computer program is a collection of instructions that completes some desired task. It is often found that the same sequence of statements is required many times in a program. Sometime a large program is divided into independent short programs. These short programs are called subroutines. In BASIC a subroutine is a series of statements that is written once in a program but it may be used a number of times anywhere in a program as and when desired.

THE GOSUB / RETURN Statement

There are two statements in BASIC that allow us to use the subroutines; the GOSUB and the RETURN statements. The GOSUB statement will cause the transfer of control to the subroutines. When a GOSUB statement is encountered, control is transferred to new line #. An example of GOSUB instruction is:-

40 GOSUB 270

Where 270 is the line # of the first statement in the subroutine.

A subroutine begins at the line number referred by the GOSUB instruction and ends with the RETURN instruction. After the subroutine has been executed, the RETURN instruction passes control back to the statement immediately following the GOSUB instruction that has called the subroutine as shown below:-

```
40    GOSUB  270
50
•
•
120
130   END
•
•
```

```
270   REM BEGINNING OF GOSUB
 .
 .          (Execute the statement)
300   RETURN
```

The GOSUB statement will cause a transfer of control to the subroutine and then RETURN will transfer THE CONTROL back to the statement following GOSUB. The last instruction in any subroutine must always be the RETURN instruction. Otherwise control will not be returned from the subroutine.

In addition to make program easier to understand, the use of subroutines can save typing time and debugging effort when used to handle tasks that occur repeatedly in a program as illustrated in the following program.

PROGRAM 13.11

```
100   FOR J = 1 TO 4
110   GOSUB 300
120   GOSUB 500
130   NEXT J
140   END
300   LET P = J * J + 5 J
310   RETURN
500   PRINT J, P
510   PRINT "CORRECT"
520   RETURN
```

OUTPUT OF PROGRAM 13.11

```
1                    6
CORRECT
2                    14
CORRECT
3                    24
CORRECT
4                    36
CORRECT
```

Note that END statement must be placed at the end of the main program before the subroutines; otherwise, the computer will simply start executing the first subroutine after finishing the main program.

13.8 FILE HANDLING

A computer can process large amounts of data. It is always useful to store input and output in a data file. A file consists of record stored on a secondary storage device such as disk. These records can be organized in different ways. Using file handling techniques, the same data can be used over and over again or modified. BASIC supports two different techniques to organize files. These are:-

- Sequential files and
- Random access files

THE OPEN AND CLOSE STATEMENTS

Before a file can be read or written, it must first be activated or accessed with the use of an OPEN statement. The OPEN and CLOSE statements are used to open file for processing and close it after necessary processing.

An OPEN statement must contain a reference to the file. For example:-

80 OPEN "DATAFILE" FOR OUTPUT AS # 1

This OPEN statement opens a file named DATAFILE to be used for output. The computer will refer to this as file number 1, and any further input from or output to this file must use this file number.

To be used for input, a file must already exist. The OPEN statement shown above will prepare the file named DATAFILE to receive output by a program. Later, we can use the same file for input another program. Once a file has been opened for output, we can write information to it using the WRITE statement, which is like the PRINT statement except that it places a comma after each item of the output. Because commas must separate data items in the input, that we place on a file via the WRITE statement and can also be used as input to another file. The general form of the WRITE statement is:-

Line # WRITE # n, *variable list*

Where n = 1, 2, 3, …, indicating the file number into which data is entered. Suppose we want to place the following record on DATAFILE:

Students Name "FATIMA", Age 12 and Sex "F"

Following program statements accomplish this after we open DATAFILE as file #1:-

```
80    STUDENT$   =   "FATIMA"
90    AGE   =   12
100   SEX$   =   "F"
110   WRITE #1, STUDENT$, AGE, SEX$
120   CLOSE #1
```

The WRITE #1 statement outputs the information to the file called DATAFILE, which the OPEN statement has designated it as file number 1. After entering all the record into the file, we close it with the CLOSE statement.

A complete program to create DATAFILE using keyboard input is shown as Program 13.12. We use a FOR loop for a given number of records to input the data and then to output it to the file.

PROGRAM 13.12

```
200    REM** PROGRAM TO SET UP FILE WITH STUDENT DATA**
210    GOSUB 400 'SET UP FILE
220    GOSUB 600 'INPUT DATA AND WRITE TO FILE
230    END
400    REM*****SUBROUTINE TO SET UP FILE******
410    OPEN "DATAFILE" FOR OUTPUT AS #1
420    INPUT "ENTER THE NUMBER OF RECORDS: "; NUM. RECS
430    WRITE #1, NUM. RECS 'FIRST RECORD ON FILE IS NUMBER OF STUDENTS
440    RETURN
600    REM ** SUBROUTINE TO INPUT DATA AND THEN  WRITE TO DATAFILE **
610    FOR I = 1 TO NUM. RECS 'USE INPUT / OUTPUT LOOP
620    INPUT "ENTER NAME, AGE, AND SEX: "; STUDENT$, AGE, SEX$
630    WRITE ≠ 1, STUDENT$, AGE, SEX$ 'OUTPUT RECORD TO FILE
640    NEXT I 'END OF INPUT / OUTPUT LOOP
650    CLOSE #1 'CLOSE THE FILE
660    RETURN
```

If Program 13.12 is executed for five records, the Output would appear as:-

PROGRAM 13.12 EXECUTION RESULTS

```
ENTER THE NUMBER OF RECORDS:? 5
ENTER NAME, AGE, AND SEX:          ? NEELUM SULTAN, 18, F
ENTER NAME, AGE, AND SEX:          ? NUDRAT KHALID, 19, F
ENTER NAME, AGE, AND SEX:          ? BILAL ASLAM, 20, M
ENTER NAME, AGE, AND SEX:          ? NAUMAN LODHI, 18, M
ENTER NAME, AGE, AND SEX:          ? RABIA SAQIB, 25, F
```

If we now wanted to input data from DATAFILE and display it on the screen, we would open the file for input and use the INPUT #n and PRINT statements to display data from the file. The INPUT #n statement has the same form in Program 13.13 as the WRITE #n statement in Program 13.12, where the number of records on DATAFILE is input first and then the records from the file.

PROGRAM 13.13

```
100    REM ** PROGRAM TO INPUT STUDENT DATA FROM A FILE **
110    GOSUB 200 'OPEN FILE AND INUPT NUMBER OF RECORDS
120    GOSUB 400 'INPUT STUDENT DATA FROM FILE
130    END
200    REM *SUBROUTINE TO OPEN FILE AND INPUT NUMBER OF RECORDS*
210    OPEN "DATAFILE" FOR INPUT AS #2 'OPEN FILE FOR INPUT
220    INPUT #2, NUMBER 'READ NUMBER OF STUDENT RECORDS ON FILE
230    RETURN
400    REM **SUBROUTINE TO INPUT AND OUTPUT STUDENT RECORDS**
410    FOR N = 1 TO NUMBER 'USE INPUT / OUTPUT LOOP
620    INPUT #2, STUDENT, NAMES$, YEARS, MALE, FEMALE$ ' INPUT DATA
630    PRINT "NAME: "; STUDENT. NAME$;" AGE: " ; YEARS;" SEX; "; MALE. FEMALE$
640    NEXT N 'END OF INPUT / OUTPUT LOOP
650    CLOSE #2 'CLOSE THE FILE
660    RETURN
```

PROGRAM 13.13 EXECUTION RESULTS

NAME:	NEELUM SULTAN,	AGE:	18	SEX:	F
NAME:	NUDRAT KHALID,	AGE:	19	SEX:	F
NAME:	BILAL ASLAM,	AGE:	20	SEX:	M
NAME:	NAUMAN LODHI,	AGE:	18	SEX:	M
NAME:	RABIA SAQIB,	AGE:	25	SEX:	F

For Labs. consult **Computer Lab. Journal** for SSC. Computer Science by the author.

SUMMARY

A structured program is written in modules or blocks and allows only one entrance to and only one exit from each module .

SUB-PROGRAMS

The primary idea of modular programming is to break up large problem into smaller, manageable parts called **modules** or **subprograms**. A subprogram is a group of statements that performs a single operation on execution. Subprograms are written once anywhere in a main program but may be called at several places.

FUNCTIONS

A function is a routine that takes a value (string or number) as an input, and produces a single value as output. These are two types of functions:-

System-defined functions/built-in-functions and user-defined functions.

SYSTEM DEFINED FUNCTIONS/BUILT-IN-FUNCTIONS

BASIC includes a wide range of built-in-functions. These built-in-functions are also called library functions or system's defined functions.

NUMERIC FUNCTIONS

These include ABS, INT, FIX, RND, SQR, LOG, SIN, COS, TAN etc. *These functions can be executed in the direct mode and in programming mode.*

ABS function produces the absolute value of an expression.

CINT function rounds the fractional part of an expression.

INT function prints the greatest integer less than or equal an expression.

RND function It generates a number between 0 and 1

SGN function reports the SIGN of a number.

FIX function gives an integer value by simply dropping off the fractional part.

SQR function is used to find the square root of any positive number.

LOG function is used to find the value of natural logarithm.

SIN function is used to find the sine of an angle expressed in radians.

COS function is used to find cosine of an angle expressed in radians.
TAN function is used to find the tangent of an angle expressed in radians.

STRING FUNCTIONS These include HEX$, LEFT$, MID$, RIGHT$, CHR$, STR$, TIME$, INKEY$ etc.

HEX$ function calculates Hexadecimal number of given decimal number
LEFT$ function is used to select n left most characters of a given string
RIGHT$ function selects rightmost " n " characters of a given string.
INKEY$ function responds to any key without pressing the ENTER key.
SPACE function leaves or skips certain number of spaces before a character.

USER DEFINED FUNCTIONS

Besides providing the built-in-functions, BASIC has the statement DEF FN that lets you to define your own functions. These are called user-defined functions. Once a function is defined, it may be used anywhere in the program in the same way in which built-in-functions are used. Moreover a user-defined function can easily be changed to another function.

SUBROUTINE SUBPROGRAMS

It is often found that the same sequence of statements is required many times in a program. These short programs are called subroutines.

THE GOSUB / RETURN Statement

In BASIC, GOSUB and the RETURN statements allow the use of subroutines. When a GOSUB statement is encountered, control is transferred to new line #. A subroutine begins at the line number referred by the GOSUB instruction and ends with the RETURN instruction. The RETURN instruction passes control back to the statement immediately after GOSUB instruction that has called the subroutine

FILE HANDLING

It is always useful to store input and output in a data file. These records can be organized in different ways. Using file handling techniques, the same data can be used over and over again.

THE OPEN AND CLOSE STATEMENTS

Before a file can be read or written, it must first be activated or accessed with the use of an OPEN statement. The OPEN and CLOSE statements are used to open file for processing and to close it after necessary processing.

To be used for input, a file must already exist. Once a file has been opened for output, we can write information on it using the WRITE statement, which is like the PRINT statement except that it places a comma after each item of the output.

EXERCISE

13.01 Complete the following statements

i) A subprogram is a part of a program which is written __________ in the main program but is used at many places in the program.

ii) Subroutines are used to duplication and ____________ time.

iii) Subroutines are used in BASIC to break up program into ___________.

iv) A GOSUB statement must end with a ___________ statement.

v) An __________ statement in used when we want to read or write a file.

13.02 Tick (✓) the following statements either True or False.

i) The built-in-functions are also called library functions T/F

ii) A function is also called subroutine T/F

iii) A subroutine is an independent part of the program that performs a specific task. T/F

iv) End statement must be placed at the end of a subroutine. T/F

v) BASIC supports two different techniques to organize files. T/F

13.03 Encircle one choice A, B, C or D in each case.

i) On execution of PRINT FIX (-6.83), the output will be

(A) 6 (B) 6.83 (C) - 6 (D) - 7

ii) The result of execution of PRINT ABS (-4.58)

(A) 4 (B) - 4 (C) 5 (D) 4.58

iii) In BASIC LOG function refers to natural logarithm having Base.

(A) 2 (B) 10 (C) 5 (D) e

iv) LEFT (A$,n) is used to:-

(A) leave n spaces before the strung A$

(B) Leave n spaces before every character of the strung A $

(C) Leave n left characters of the strung A$

(D) Select n characters of the strung A$

v) RETURN Statement passes control back to:

(A) GOSUB statement (B) Next to BOSUB statement

(C) Beginning of the program (D) The statement before GOSUB.

13.04 Match the items given in Column I with those in Column II

Column I	Column II
i) ASCII CODE	a) LIBRARY FUNCTION
ii) FILE	b) LOG e
iii) GO SUB	c) CHR$
iv) BUILT-IN-FUNCTION	d) OPEN
v) 1	e) RETURN
	f) END

13.05 What are subprograms? Name the type of subprograms? What is the advantage of subprograms?

13.06 Explain two types of subprograms? How do they differ from each other?

13.07 Define the function and their types.

13.08 What is system defined function? What is difference between system defined function and user defined functions?

13.09 Describe the following library functions:

ABS, CINT, HEX, INT, FIX, OCT, RND, SGN, CHR$, INKEY$,

13.10 What would be displayed by each of the following PRINT statement?

a) PRINT SGN (22.74)
b) PRINT INT (-8.8)
c) PRINT ABS (-17.53)
d) PRINT FIX (- 17.53)
e) PRINT SQR (25)
f) PRINT LOG (1)
g) PRINT SIN (π)
h) PRINT COS (π/2)
i) PRINT TAN (0)
j) PRINT RND (5.39)

13.11 What is the purpose of users defined functions? Explain with example.

13.12 What is a subroutine subprogram? When and where it can be used in a program?

13.13 What is the purpose of GOSUB/RETURN statements? How it differs from GOTO statement can a subroutine call another subroutine?

13.14 Find the error in the following programs:

a)
```
10  DEF FNI (X) = X^ 3 + 1
20  INPUT P
30  PRINT FNI(P)
40  END
```

b)
```
10   GOSUB 150
20   GOSUB 200
150  PRINT " BOOK"
160  RETURN
170  PRINT "SUNDAY"
180  RETURN
```

13.15 Translate into $\sqrt{16+x^2}$ DEF FN statement.

13.16 Write a program using DEF FN statement to convert the temperature from Celsius (C) to Fahrenheit (F), such that $F = \frac{9}{5}C + 32$

13.17 Write a main routine that uses subroutines 200 and 400 to PRINT a block letter E and T as shown.

```
XXXXXX     MMMMMMM
X             M
X             M
XXXXXX        M
X             M
X             M
XXXXXX
```

13.18 What are file? How files are handled in computer?

13.19 How a file can be opened and closed?

13.20 How to read and write into a file?

14

GRAPHICS IN BASIC

On most micro-computers, it is possible to produce graphics output in addition to printing letters, numbers and other usual characters, the computer can draw fine **dots** called **picture elements** or **pixels** on the screen at any locations you specify. It is hard to see and measure a single pixel. When a line is drawn, computer places a string of pixels adjacent to each other. By writing a program that plots many pixels, you can make the computer draw design or picture.

The size of pixels and colors available also vary from computer to computer. Different kinds of graphs and drawings can be produced easily using graphic commands. For example using instructions such as LINE, PSET, DRAW, CIRCLE you can produce triangles, rectangles, stars, circles etc. For Labs. consult **Computer Lab. Journal** for SSC. Computer Science by Prof.. M. Tahir Hassan.

14.1 SCREEN FUNCTION

Normally the monitor screen is in the text mode. In this mode we cannot draw a good picture. For obtaining a good picture the number of location i.e rows and column must be increased. This is done either by immediate mode command or a statement in a program to obtain medium resolution graphic or high resolution graphic. The format of the SCREEN statement is:-

Line # SCREEN *mode*

The screen function can be set to either of the following three modes.

- Text mode. (SCREEN 0)
- Medium resolution graphics mode. (SCREEN 1)
- High resolution graphics mode. (SCREEN2)

TEXT MODE

In the text mode characters are displayed on the screen in 40 columns and 25 rows (40 × 25) or 80 columns and 25 rows (80 × 25) depending upon the selection of screen width.

MEDIUM RESOLUTION GRAPHICS MODE

In medium resolution graphic mode, the screen is divided into a matrix consisting of 320×200 points with 4 COLORS i.e. the display screen is divided into 320 columns and 200 rows of pixels. The individual points on the screen are referenced by specifying the columns first and then row. Columns are numbered from 0 to 319 and rows from 0 to 199. For example (0, 0) is referring to the left upper corner and (319, 199) is used to refer to the lower right corner. The four colors are numbered 0, 1, 2, 3. You can choose any of the 16 colors as background and one of the color set as foreground color.

HIGH RESOLUTION GRAPHICS MODE

In high resolution mode, the screen is divided into 640×200 points in black and white i.e. 640 columns and 200 rows of pixels, which permits more precise drawings. Columns are numbered from 0 to 639 and rows from 0 to 199. (0, 0) indicates the upper left corner and (639, 199) indicates the lower right corner. Text characters can also be displayed in high resolution graphic mode, where there are 25 lines of 80 characters each.

14.2 COLOR STATEMENT

The screen statement does not clear the screen (except it is used first time) so to clear the screen use CLS when color monitor is used. Next is to tell the computer about the background colors you want for the screen using the COLOR statement. The format of the COLOR statement depends on the screen mode. COLOR will not function in high resolution graphic mode i.e., in SCREEN 2. In the **text mode** (SCREEN 0), the formate of the statement will be:-

Line # COLOR [*foreground*], [*background*]

The foreground color number must be an integer from 0 to 31. The color numbers 16 to 31 are the blinking versions of the colors from 0 to 15 given in table 14.1(a) and are found by adding 16 into the corresponding color number from 0 to 15. For example if foreground color number is 2 (for green), then blinking green will be 18 (i.e., 2 +16)

The background color must be an integer from 0 to 7 given in the table 17.1(a). Note that if the background color is the same as that of the foreground, then the characters displayed on the screen will not be visible. (Blinking colors in background are not permitted). Consider the following statements:-

```
20   SCREEN 0
30   COLOR 13, 2, 4
```

Statement in line # 20 sets the screen in the text mode, and the statement in line # 30 sets the text light magenta in green background.

In the **medium resolution graphic** mode SCREEN 1, the color statement has the following format:-

Line # COLOR [*background*], [*palette*]

16 background colors are available having code numbers from 0 to 15 as given in the table 14.1(a). After selecting the background color of the screen, select one of two palettes (a palette is the range of color used by an artist in a painting), each contain four possible colors for the points that will be drawn on the back ground. The palettes are numbered 0 and 1 each with four colors as given in table 14.2 (b). and will be selected by the color parameter of LINE, CIRCLE, PSET, DRAW or other graphic statements.

COLOUR	CODE	COLOUR	CODE
Black	0	Gray	8
Blue	1	Light Blue	9
Green	2	Light Green	10
Cyan	3	Light Cyan	11
Red	4	Light Red	12
Magenta	5	Light Magenta	13
Brown	6	Yellow	14
White	7	Bright White	15

	COLOUR	CODE
PALETTE 0	Background	0
	Green	1
	Red	2
	Brown	3
PALETTE 1	Background	0
	Cyan	1
	Magenta	2
	White	3

Table 14.1 (a) Background color and their code (b) Palettes with foreground color code

Consider the following statement in the medium resolution mode:-

```
10  SCREEN 1
20  COLOR 7, 0
30  PSET (50,50), 2
```

will set the screen in medium resolution graphic mode and the background color to white with a point in red at location 50,50 on the screen of 320×200 matrix.

14.3 MEDIUM RESOLUTION GRAPHICS

PSET statement is used to specify the coordinates X, Y of the point, where it appears on the screen. The following program illustrate a red and white letter on a screen with blue background

PROGRAM 14.1

```
10  REM ** RED VERTICAL LINE OF 5 PIXELS**
15  CLS
20  SCREEN  1,0
30  COLOR   1,0
40  PSET    (40, 10), 2
50  PSET    (40, 11), 2
60  PSET    (40, 12), 2
70  PSET    (40, 13), 2
```

```
80   PSET    (40, 14), 2
90   REM** WHITE HORIZONTAL LINE OF HEIGHT 4 PIXELS**
100  COLOR   1,1
110  PSET    (41, 14), 3
120  PSET    (42, 14), 3
130  PSET    (43, 14), 3
140  PSET    (44, 14), 3
150  END
```

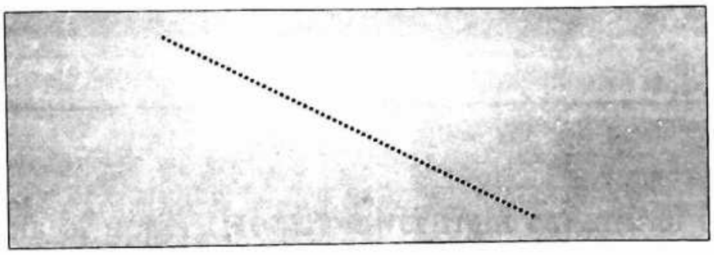
FOR ... NEXT loop can be used to draw a horizontal or vertical line using medium resolution graphic as illustrated below.

PROGRAM 14.2

```
10   REM ** TO DRAW A LINE USING FOR ... NEXT LOOP **
20   SCREEN  1,0
30   COLOR   2,0
40   FOR     A = 0 TO 60
50   X = 80 + 2*A
60   Y = 50 + A
70   PSET    (X, Y),2
80   NEXT    A
90   END
```

OUTPUT OF PROGRAM 14.2

A red Horizontal line from column 80, row 50 to column 200, row 110.



THE LINE STATEMENT

A line connecting any two point can be drawn using LINE statement. The syntax of LINE is:-

LINE (*X1, Y1*) - (X2, Y2), *Color*

Thus program 14.2 can be rewritten using LINE statement for the same output.

PROGRAM 14.3

```
10   REM ** DRAW A LINE USING LINE COMMAND**
20   SCREEN  1,0
30   COLOR   2,0
40   LINE    (80, 50) - (200, 110), 2
50   END
```

Note that line # 40 with LINE statement in program 14.3 has replaced the lines # 40 to 80 using PSET statement in program 14.2.

14.4 HIGH RESOLUTION GRAPHICS

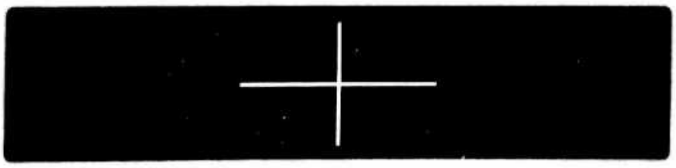
High resolution graphics mode allows to draw more finely detailed shapes than those possible on medium resolution graphics. This is because the number of rows and columns are twice the number available in medium resolution. A slightly different screen instruction must be given to do high resolution graphics. The instruction SCREEN 2 (instead of SCREEN 1,0) either on immediate mode or program statement gets up the screen for high resolution graphics.

The PSET and LINE statements are used to draw points and lines in high resolution graphics. These statements have the same form as in medium resolution graphics except that the number 1 at the end of either statement indicates a white point or line and the number 0, a black point or line. However a black point is invisible on a black screen. The following program illustrate to draw a plus symbol in high resolution graphics.

PROGRAM 14.4

```
10    REM ** PLUS SYMBOL **
10    CLS
20    SCREEN 2
30    LINE (100,100) - (300,100), 1
40    LINE (200,50) - (200,150), 1
50    END
```

OUTPUT OF PROGRAM 14.4

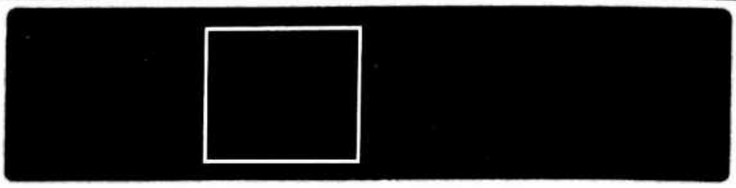


The following program illustrate to draw a rectangle.

PROGRAM 14. 5

```
10    REM ** HI-RES RECTANGLE **
10    CLS
20    SCREEN 2
30    LINE (40,40) - (160, 40), 1
40    LINE - (160, 120), 1
50    LINE - (40, 120), 1
60    LINE - (40, 40), 1
70    END
```

OUTPUT OF PROGRAM 14.5



The output will be a rectangle with horizontal sides from (40, 40) to (160,40) and (40, 120) to (160, 140) and vertical sides from (40, 40) to (40, 120) and (160,40) to (160, 120) as shown in the output of program 14.6.

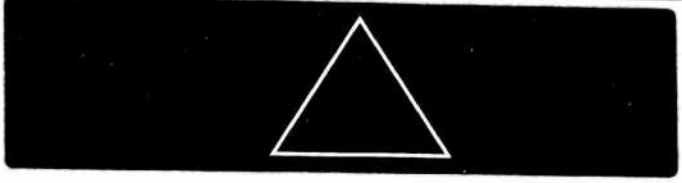
The following program 14.7. illustrates to draw a triangle in high resolution graphic mode.

PROGRAM 14.6

```
10      REM * HI-RES TRIANGLE
20      CLS
30      SCREEN          2
40      LINE    (100, 40) - (150,90), 1
50      LINE    - (50, 90), 1
60      LINE    - (100, 40), 1
70      END
```

A white line from coordinates (100, 40) joined with coordinate (150, 90), second white line from coordinate (150, 90) to (50, 90) and the third white line from coordinately (50, 90) to (100, 40) form a triangle as shown.

OUTPUT OF PROGRAM 14.6



Note that in program lines 50 and 60, first pair of coordinates is missing in a LINE statement, then the line is drawn from the 1st point plotted (by either PSET or LINE) to the point immediately after the dash.

Rectangle or square box can be drawn by adding B using line statement with points of upper left and lower right corners of the square or rectangle. The coordinates of these points are (X1, Y1) and (X2, Y2) as shown in figure 14.1

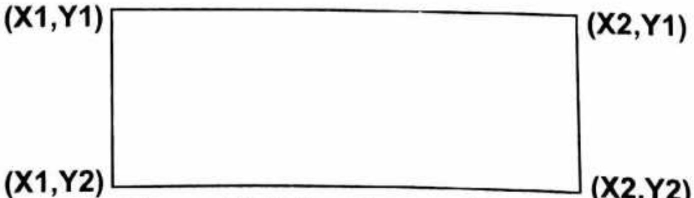


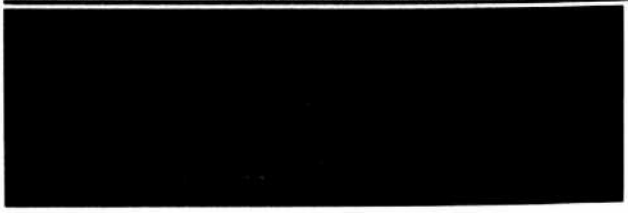
Figure 14.1 Coordinates of a box

The following program draws a square box of each side 50 pixels inside a bigger square of side 60 pixels:-

PROGRAM 14.7

```
10      REM ** RED SQUARE BOX INSIDE A GREEN BOX**
20      SCREEN 1
30      LINE    (50, 50) - (110,110), 1, B
40      LINE    (55, 55) - (105,105), 2, B
```

OUTPUT OF PROGRAM 14.7



14.5 DRAW STATEMENT

Draw is a powerful graphic program. It is used to draw lines and other shapes. DRAW statement has the following format:-

Line # DRAW *string*

A string consists of single character command followed by a prefix that controls the size, direction etc. of the line and is enclosed in the quotation marks. For example:-

50 DRAW " E30"

The drawing commands that may be used are given below with the digit "n" as the prefix to describe the number of points moved by the command.

COMMAND		PURPOSE
↑	Un	Upward move through n points.
↓	Dn	Downward move through n points.
→	Rn	Rightward move through n points.
←	Ln	Leftward move through n points.
↗	En	Diagonally right upward move through n points.
↘	Fn	Diagonally right downward move through n points.
↙	Gn	Diagonally left downward move through n points.
↖	Hn	Diagonally left upward move through n points.

On execution of a command, a line is drawn from the point last referred. The number of points and the direction is specified by the drawing command followed by prefix.

- B Move without leaving any mark.
- N Move to starting position after finish.

Following programs illustrate the applications of DRAW statement used to draw various drawing objects.

PROGRAM 14.8

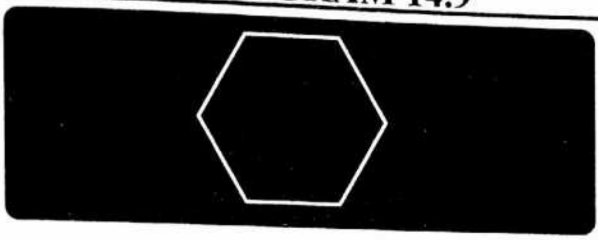
```
10   REM ** RECTANGLE **
20   SCREEN 1
30   DRAW "U80 R80 D 80 L80"
```

OUTPUT OF PROGRAM 14.8

PROGRAM 14.9

```
10     REM ** CONSTRUCTING REGULAR HEXAGON**
20     SCREEN 1
30     DRAW "E30 R50 F30 G30 L50 H30"
```

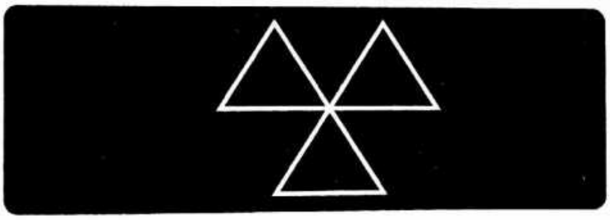
OUTPUT OF PROGRAM 14.9



PROGRAM 14.10

```
10     REM ** THREE SYMMETRICAL TRIANGLES WITH ONE COMMON CORNER **
20      CLS
30     SCREEN 1
40     DRAW "E30 F60  L60 E60  F30 L120"
```

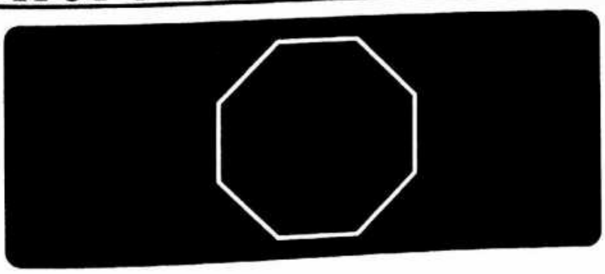
OUTPUT OF PROGRAM 14.10



PROGRAM 14.11

```
10     REM **  CONSTRUCTING REGULAR OCTAGON **
20     SCREEN 2
30     DRAW " E20 R30 F20 D25 G20 L30 H20 U25 "
```

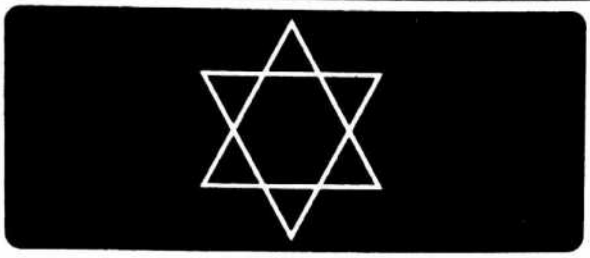
OUTPUT OF PROGRAM 14.11



PROGRAM 14.12

```
10    REM ** SIX CORNERS STAR**
20    SCREEN 2
40    DRAW " E60 F60 L120BU40 R120G60 H60"
```

OUTPUT OF PROGRAM 14.12



14.6 CIRCLES AND ELLIPSES

In BASIC, there is a built-in command that will draw circles, part of a circle or ellipse. In medium resolution graphics it has the following format:-

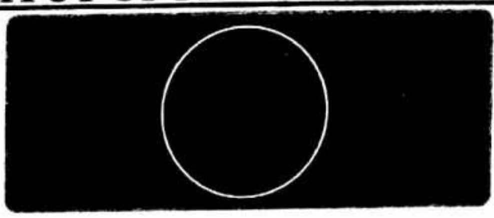
CIRCLE (X,Y), R , C , S , E , A

It will draw a circle of radius R with center at (X, Y) in the color C (selected from the palette previously chosen in the COLOR statement), S is start arc position , E is the end arc position, and A is the aspect ratio (height/width). The program 14.14 illustrate to draw a circle.

PROGRAM 14.13

```
10    REM     **DRAWING A CIRCLE OF RADIUS 50 PIXELS**
20    CLS
30    SCREEN        1,0
40    COLOR 2,0
50    CIRCLE (150, 100),50,2
60    END
```

OUTPUT OF PROGRAM 14.13

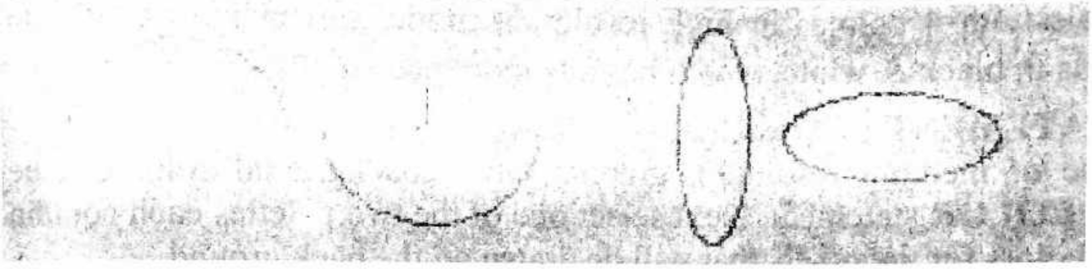


The following program illustrates upper semicircle of radius R as 30, with C as 3 for brown color from starting angle S as 0 to ending angle E as π in statement on line # 50. Similarly line # 60 produces a lower semicircle of radius 30 in green color from to 0 . the program in line # 80 and 90 taking aspect ratio A 3 and 0.4 produce vertical and horizontal ellipses respectively in red and green colors with white background in medium resolution graphic mode.

PROGRAM 14.14

```
 10  REM** BROWN UPPER  AND GREEN LOWER SEMICIRCLES **
 20  CLS
 30  SCREEN 1, 0
 40  COLOR 7, 0
 50  CIRCLE (50, 100), 30, 3, 0,3.14
 60  CIRCLE (120, 100), 30, 1,3.14
 70  REM** RED VERTICLE AND GREEN HORIZONTAL ELLIPSES**
 80  CIRCLE (200, 100) ,30, 2,,,3
 90  CIRCLE (250, 100) ,30, 1,,,0.4
100  END
```

OUTPUT PROGRAM 14.14

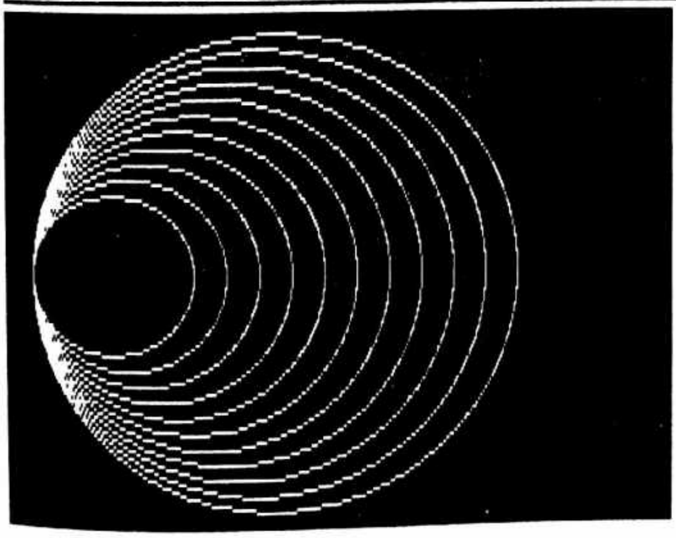


The following program illustrates to draw 11 coinciding circles of different radii.

PROGRAM 14.15

```
10  REM** COINCIDING CIRCLES **
20  CLS
30  SCREEN 2
40  FOR  A  =  50 TO 150 STEP 10
50  B = A + 50
50  CIRCLE (B, 100), A
60  NEXT A
70  END
```

OUTPUT OF PROGRAM 14.15



SUMMARY

A picture is composed of fine **dots** called **picture elements** or **pixels.** A design or picture can be drawn through a computer program consisting of graphic commands that plots many pixels.

SCREEN FUNCTION

The screen function can be set to any of the following three modes. These are Text mode, Medium resolution graphics mode and High resolution graphics mode. In the text mode there are generally 80 columns and 25 rows. In medium resolution graphic mode, the screen has a matrix of 320 columns and 200 rows or 320×200 pixels with 4 colors. In high resolution mode, screen is divided into 640×200 pixels in black & white, which permits more precise drawings.

COLOR STATEMENT

In case of medium resolution graphic mode, background color can be selected from **COLOR** statement. Next select one of the two palettes each contain four possiblé colors for the pixels that will be drawn on the back ground.

PSET statement

PSET is used to specify the coordinates X, Y of a point on the screen.

LINE statement

A line can be drawn between any two point having coordinates (X1, Y1) and (X2, Y2) using LINE statement.

DRAW STATEMENT

Draw is a powerful graphic program. It is used to draw lines and other shapes. DRAW statement is followed by a prefix that controls the size, direction etc. of the line. The prefix to move *n* points upward is **U, D** for Downward, R for Rightward, **L** for Leftward, **E** for Diagonally right upward, **F** for Diagonally right downward, **G** for Diagonally left downward, **H** for Diagonally left upward. Command **B** is used for a move without leaving any mark and **N** for a move to starting position after finish.

CIRCLES AND ELLIPSES

In BASIC, there is a built-in command that will draw circles, part of a circle or ellipse. In medium resolution graphics it has the following format:-

CIRCLE (X,Y), R , C , S , E , A

It will draw a circle of radius R with center at (X, Y) in color C (selected from the palette previously chosen in the COLOR statement), S is start and E is the end arc positions and A is the aspect ratio (height/width).

EXERCISE

14.01 Complete the following statements:

i) The location which is the intersection of a row and a column on a screen is called ____________.

ii) In high resolution the screen is divided into a matrix of ______ pixels.

iii) In LINE statement line starts from ______________ when initial coordinate are missing.

iv) U 50 means ____________________________ in Draw statement.

v) In DRAW command the prefix command _______move to starting position after finish.

14.02 Tick (✓) the following statements either True or False.

i) In medium resolution the screen is divided into a matrix of 640×200 pixels — True/False

ii) In medium resolution graphic mode the color code 20 corresponds to the same color as for code 4. — True/False

iii) COLOR will not function in High resolution graphic mode. True/False

iv) In Draw statement, the command F 80 move diagonally left down ward through 80 pixel. — True/False

v) The circle statement can also be used to draw an ellipse — True/False

14.03 Encircle one choice A, B, C or D in each case.

i) The statement PSET (50, 150) will draw a point on the screen at the:-
(A) Left Top (B) Right Top (C) Left Bottom (D) Centre

ii) The color of code 2 belonging to palette 1 is:-
(A) Cyan (B) White (C) Magenta (D) Red

iii) In Draw statement B is used for:-
(A) Back movement (B) Blank movement
(C) Vertical movement (D) Diagonal movement

iv) Number of circles drawn on execution by the following statements are:-

```
10  SCREEN 2
20  FOR R = 40 TO 120 STEP 20
30  CIRCLE (150, 100), R
40  END
```

(A) 1 (B) 4 (C) 5 (D) None

v) The variable that enables to draw ellipse in CIRCLE statement is the:-
(A) Fourth (B) Fifth (C) Sixth (D) Seventh

14.04 Match the items given in Column I with those in Column II

Column I		Column II	
i)	PSET	a)	(50, 120),30,2,,,2
ii)	No line	b)	E
iii)	CIRCLE	c)	N
iv)	Diagonal line	d)	(100, 50)
v)	Vertical Ellipse	e)	(50, 100),30
		f)	B

14.05 What is meant by pixel and resolution?

14.06 Describe the different modes to display graphic on the screen.

14.07 What do you mean by background color and palettes?

14.08 What is the purpose and syntax of the PSET statement? Give examples.

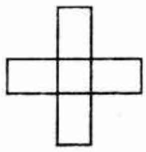
14.09 What is the purpose and syntax of the line statement? Give examples.

14.10 Predict the output of the following

a)
```
10    CLS
20    SCREEN  1, 0
30    COLOR   0, 1
40    LINE    (20, 20)-(45, 20), 1
50    END
```

b)
```
10    CLS
20    SCREEN  2
30    LINE  (20, 20)-(100, 20),1
40    LINE  - (100, 100),1
50    LINE  - (20,100),1
60    LINE  - (20, 20),1
70    END
```

14.11 Write a program to draw the pattern shown in Figure (a) and (b).



(a)

(b)

14.12 What is the purpose and syntax of Draw statement? Which commands in DRAW statement are used for diagonal movement?

14.13 What is the purpose of CIRCLE statement? What is its syntax?

14.14 What is the syntax when only a part of a circle is to be drawn? How can CIRCLE statement is used to generate an ellipse?

14.15 Write a program in high resolution graphics to draw the word **EXIT** in large block letters on the screen.

14.16 RUN the program 14.14 after changing the line number 50 by CIRCLE (150, 100), 50,2,0,1 and describe the output.

14.17 Try this instruction in the program 14.14 in line # 50 by using CIRCLE (150, 100), 50,2,1,0.5, What does it do?

14.18 Write a program to produce five concentric circles of different radii.

15

WORD PROCESSING

Word processing refers to the use of a computer program to prepare and print documents. Through this software, a user enters the text with the help of input devices, views it on the monitor, changes it as necessary, saves the document on disk, and gets a hard copy of it from the printer.

Other useful functions of word processing software are setting of margins, line spacing, automatically wrapping the text around the margins inserting, deleting, and searching the document for a particular sequence of characters, underline, boldface or italicized material. An important operation with word processing software is the ability to electrically "cut" or "copy" a block of material from a document and "paste" it any number of times, somewhere else either in the same or some other document. For Labs. on MS Word, consult **Computer Lab. Journal** for SSC. Computer Science by Prof. M. Tahir Hassan.

15.1 WORD PROCESSING PACKAGES

Software packages like Word Perfect, Word Star, and MS Word are all examples of Word Processing packages for personal computers. The four main operations of a word processing package are:-

i. Defining the form of the document,
ii. Entering a document from a keyboard,
iii. Editing (modifying) the document, and
iv. Printing the document.

15.2 MICROSOFT WORD

Microsoft Words provides facilities like book marks, auto text, auto table, links, pictures, symbols, video clips, header and footer, footnotes, fonts, borders, shading, columns, spell checker, grammar checker, auto correct, thesaurus etc. Its auto recover feature automatically saves your work after a pre-defined amount of time. In Word 2000 you may also create backup copies of all your documents.

LOADING MS WORD

As you will learn to use computer, you will find diferent ways to load MS Word such as using My computer windows Explorer or through Start menu commands such as Programs, Documents etc. However in the beginning it is better to load Microsoft Word using Programs command as shown in Figure 15.1.

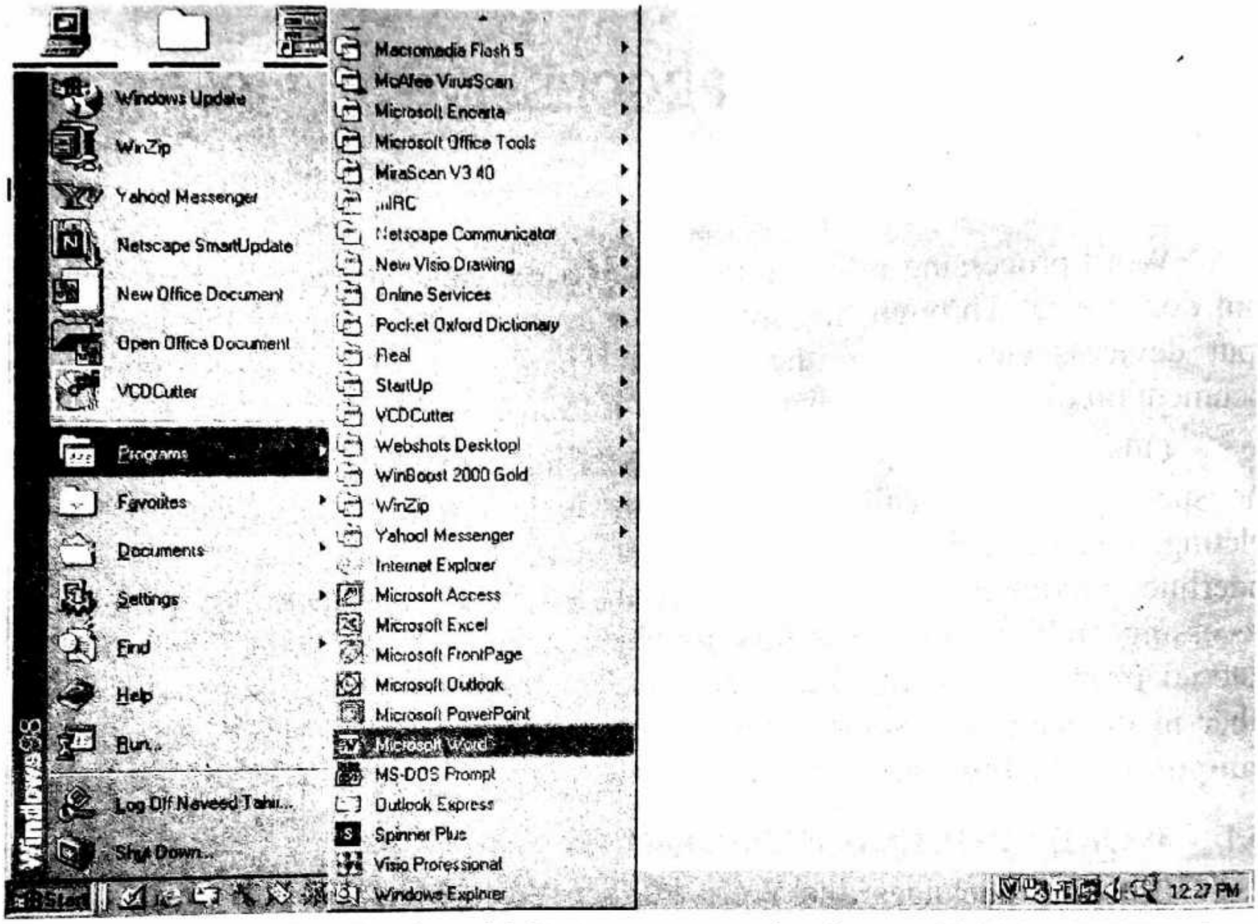


Figure 15.1 Windows Interface with **Start** menu open and ready to run highlighted **MS Word**.

Perform the following steps to load Microsoft Word.

- Point to the start button Start on the taskbar and then click the left mouse button once. The start menu appears as a pop-up menu.
- Point to the programs command using the mouse. Cascading menu displaying the list of programs appears.
- Move the mouse pointer horizontally to the right until it highlights an option in the programs menu.
- Point to the Microsoft word in the programs menu and then click the left mouse button once to execute the command. After a few seconds a blank Microsoft Word Window may appear as shown in Figure 15.2.

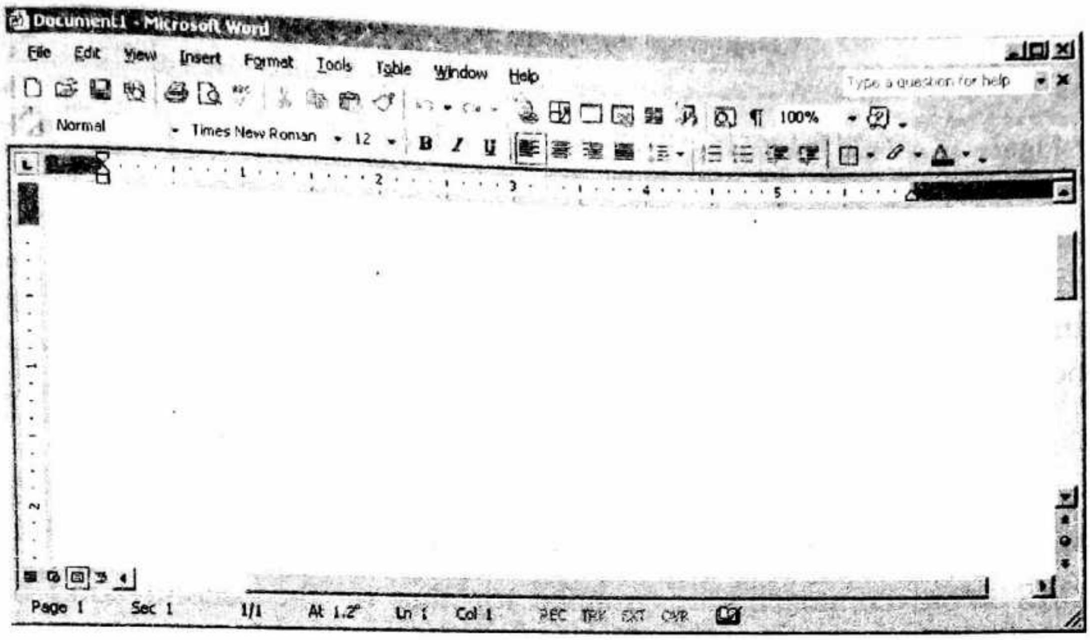


Figure 15.2 Initial **Microsoft Word** screen with no text in the document as you open it.

COMPONENTS OF MICROSOFT WORD

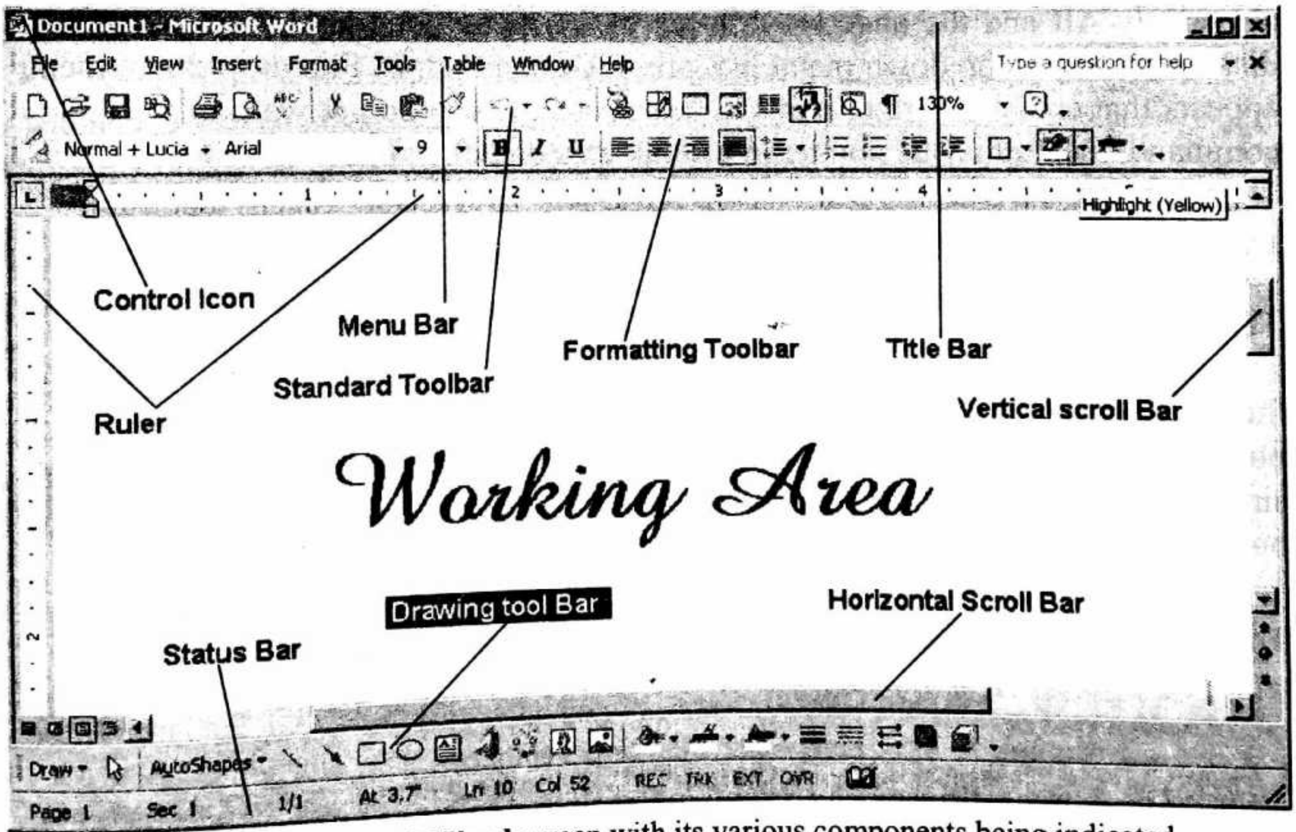


Figure 15.3 Microsoft Word screen with its various components being indicated.

Microsoft Word has the various components on its window shown in Figure 15.3 such as Title bar, Menu bar, Standard Toolbar, Formatting toolbar, Rulers, Scroll bars and the Status bar. You can add more tools from the View drop down menu.

TITLE BAR

Figure 15.4 Title Bar having (*from left*) Control Icon, Document's name and (*on the right*) Minimize, Restore and Close buttons.

It is the uppermost bar of a window having **Control Icon** on its left. Next to control icon is the name of the **program** or **document**. At the right on the title bar are **Minimize**, **Restore** and **Close** buttons. Close button marked **×** is used to close the window in a single click on the **×.**

MENU BAR

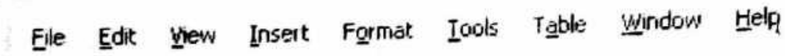


Figure 15.5 Menu bar with various command of MS Word

MS Word commands are grouped together on the second band on the window called Menu Bar as shown in the figure 15.3. Word has nine items on the Menu bar. These are **File, Edit, View, Insert, Format, Tools, Table, Window** and **Help.** To display, a command, click once on Menu bar option or press alternately **Alt** and the underlined **letter** in the menu command. For example to display the File drop down menu list, press **Alt** and then **F**. File drop down menu appears that contains further items to be selected. If you do not want to select any command, then move the pointer in a blank area and click once.

The option is not currently available if it appears '**grayed out**' within a drop down menu. A command followed by three full stops in a drop down menu list shows that on selection, a dialog box will be displayed for further options.

TOOL BARS

Assuming that you have not yet customized you word screen, you will see the **Standard** and **Formatting** toolbars appear below the Menu bar in the form of buttons. You are not required to memorize the function and button names. You simply point at any toolbar button and pause until a tool tip label appears with the button's name.

The **Standard toolbar** contains various tools as shown in Figure 15.6. These tools provide access to file management and editing commands.

Figure 15.6 Standard Toolbar

The **Formatting toolbar** contains various tools as shown in Figure 15.7. These tools are used to format character, fonts and text.

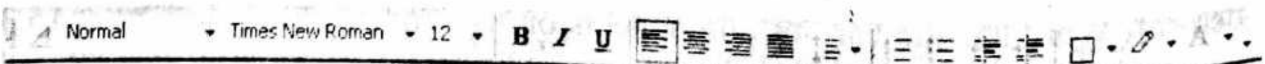


Figure 15.7 Formatting Tool Bar

THE RULER

Figure 15.8 The Ruler

Below the toolbars appears **the ruler bars, Horizontal ruler** shown in Figure 15.8 and **Vertical ruler** (not shown) are at the left of window's working area. Both these rulers are calibrated in inches.

THE HORIZOTAL SCROLL BAR

Figure 15.9 The **Horizontal Scroll Bar**

There are two **Scroll bars** used to move a document vertically and horizontally. The **Vertical bar** is on the right while the other **Horizontal Scroll bar** shown in Figure 15.9 is under the window's working area.

STATUS BAR

Figure 15.10 The **Status Bar**

Status bar is located at the bottom of the application window and above the taskbar. It displays the status or mode and other helpful information.

15.3 ENTERING TEXT INTO A WORD DOCUMENT

Microsoft Word normally functions in Insert mode but it can also function in Overtype mode.

INSERT MODE

In insert mode, text is inserted into the document at the insertion point. This is the default mode. When you type any character it appears at the insertion point while the existing text on the right moves forward to make room for the new text. Perform the following steps to insert text in your document:-

- Position the insertion point where you want to insert the new text.
- Start typing, the existing text will move to the right.

OVERTYPE MODE

In overtype mode, text overwrites the existing text from the insertion point. Perform the following steps to overwrite the existing text:-

- Position the insertion point where you want to type the new text.
- Press the **Insert** key for overtype mode. The [OVR] indicator on the **status bar** will be highlighted.
- Start typing; the new text will begin to **overwrite** on the existing text at the insertion point.

15.4 EDITING A DOCUMENT

Editing is a process making alterations in the contents of your document. Some features of editing are:-

INSERT: It is adding text in a document by placing the cursor at the desired position and start typing. The existing characters will be pushed forward.

DELETE AND UNDELETE: Deleting is removing text using Delete or Backspace keys. The unde ete (Undo) command allows you to restore the text you have deleted earli r before closing the document.

FIND AND REPLACE: The **find** and **search** command allows you to find any word, phrase, or number that exists in your document. The **Replace** command allows you to automatically replace it with something else.

CUT/COPY AND PASTE: Select (highlight) the portion of the text you want to remove or copy. Then use **cut** command to remove the selected item from the text or use **copy** command to copy the selected items without removing it. You can now paste it any number of times and anywhere in the same document or any other document. You can use clipboard especially when the number of items are more than one. Once on the clipboard, these can be pasted later on when desired.

SPELL CHECKER, GRAMMAR CHECKER AND THESAURUS: Spell checker tool checks the spelling, list correctly spelled words and provides auto correct facility to replace the typed word by the one with correct spelling. It also flags poor grammar, incomplete or too long sentences. If you are stuck for the right word, call up on screen **thesaurus** which will present you with the appropriate word or alternative words.

15.5 SAVING A DOCUMENT

When you create or edit a document, it exits only in the computer's RAM. To store the work, you must save it on a floppy, hard disk or CD. Word allows you to save your work onto the disk at any time. It is better that you save your work periodically. Because you could lose all the work you did since you last saved if something happens to the computer's power supply.

You can save a document on a disk either by save button on the standard toolbar or using save command from the File drop down menu. A dialog box appears with a suggested **filename**. You can either accept the suggested filename or type a new name. In windows you can use upto 255 characters including spaces in a filename. However the following characters are not allowed in a filename.

\ / : * ? " < > |

AUTOMATICALLY SAVING THE DOCUMENTS

By default, Word automatically saves your documents after every ten minutes. To access this feature, choose Tools and then Options … from Menu bar and select the save tab in the dialog box.

EXITING

When you finished your work, then save it before you exit. Otherwise Word asks you whether to save the changes. You can exit from the program either using the close button [X] or Exit command from File drop down menu list.

15.6 FORMATTING DOCUMENTS

Document formatting is a process of improving the over all appearance of a document. A formatted document can communicate a message more effectively than by an unformatted document. All Word Processing programs have a number of features that make it easy to format your document. This includes setting of margins and indents, fonts, column, headers and footers, borders shading, etc. Word processing programs come with default settings. Default settings are the setting automatically used by a program unless the user change them.

MS WORD'S DEFAULT SETTINGS

MS Word's default settings and Word's most commonly used character and paragraph formatting commands allow you to produce professional-looking resumes, letters, and reports with minimal effort.

Table 15.1 Setting for Normal Template

Option	Setting
Paper Size	8.5 inches wide, 11 inches height
Top and Bottom Margin	1 inch
Left and Right Margin	1.25 inch
Page Numbering	None
Line spacing	Single space
Font (Type face)	Times New Roman
Font Size	10 points
Tabs	After every 0.5 inches
Justification	Left justified with a ragged right margin

When you first load word or click the New button on the standard toolbar or click New on File Menu, a new document appears based upon Word's normal template. This template provides a set of basic setting listed in Table 15.1.

SELECTING TEXT

You can make formatting changes in a document any time before or after saving. First select the desired text and then make the change using appropriate

command. A selection may include characters, words, lines paragraph or even the entire document. Selection can be made either using a mouse as described in Table 15.2 or by keyboard short cuts. Selected text always appears highlighted in reverse (i.e. white text on black background). To remove the highlighting from the selected text after you finish formatting, press any arrow key or click anywhere in the document.

BLOCKING TEXT

MS Word can be used to select a block of text. This refers to blocking the text. Moving, clicking and dragging the mouse allow blocking of the text. MS Word provides an **invisible column** in the extreme left margin of the document window called the **selection bar**. When the mouse pointer is moved into this area, the pointer changes from I-beam to a right-pointing arrow. The selection bar also provides shortcut method for selecting a block of text using the mouse. Some of the methods of blocking text have been given in the Table 15.2.

Table 15.2 Selecting Text Using Mouse

Selected Matter	Action to be taken
Single character	Position the I-beam pointer to the left of the character that is to be selected. Press down and hold the left mouse button as you drag the mouse pointer to the right.
Single Word	Position the I-beam pointer on the word and double click the left mouse button.
Single sentence	Hold down [CTRL] and click once with the I-beam pointer positioned on any word in the sentence.
Block of text	Move the insertion point to the beginning of the block of text and then position the I-beam pointer at the end of the block. Hold down SHIFT and Click once.
Single line	Move the mouse pointer into the selection bar, beside the line to be selected. Wait until the pointer changes to a right-pointing arrow and then click once.
Single paragraph	Move the mouse pointer into the selection bar, beside the paragraph to be selected. Wait until the pointer changes to a right-pointing arrow and then double click. You can also triple click with the I-beam mouse pointer positioned inside a paragraph.
Entire Document	Move the mouse pointer into the selection bar. Wait until the pointer changes to right pointing arrow and then hold down [CTRL] and click once. You can also triple-click with the I-beam mouse pointer position in the selection bar.

DELETING BLOCKS OF TEXT

You have already learned to delete text one character at a time using the Back Space and Delete keys. However larger blocks of text can easily be deleted after making selection of the words, lines, paragraphs, or the entire document.

Perform the following steps to practice deleting and replacing text in a document.

1. Type the following document in New File.

> Computer ethics
>
> The Information age and the age of Humanity
>
> The widespread use of computers, information systems and telecommunication system has created major concern on the invasion privacy of individuals as well as of government and business organization, which lead to crimes and frauds. Computer have made data both easier to obtain and easier to store. The increased ease of obtaining data tempts organizations to collect and store more date then necessary. The control to access data becomes less when it becomes part of a huge database and its security becomes a problem.
>
> If users (individuals as well as commercial and government agencies) do not use computer data in a responsible manner and if they use information stored in the computers without the regard of others, then more laws will be passed to restrict the use of computers and computer data. Such restrictions can be very detrimental to the free flow of information. Education is a means of changing people's attitude towards security. People must come to understand that unauthorized intrusion, however innocent it may seem, is a crime. They should realize their responsibility for computer security. It is important for our society to educate people at an early age about ethical issues related to the use of computers in order to protect the free flow of information that we currently enjoy.
>
> Ethics is concerned with the standards of moral conduct. People who are unethical do not necessarily break the law. But they are engaged in activities that are damaging to others and to society at large. Computer offer us a wide range of exciting and innovative applications that can improve our lives. At the same time they also introduce ethical issues that have not been dealt with before. As a society, we have to decide if we will use this new tool for the betterment of mankind or for destructive purposes.

2. To save the document, **CLICK**: File menu, and select **Save As...** option. Type the filename and then Click save button or save the document in the suggested filename "Computer ethics" as shown above.

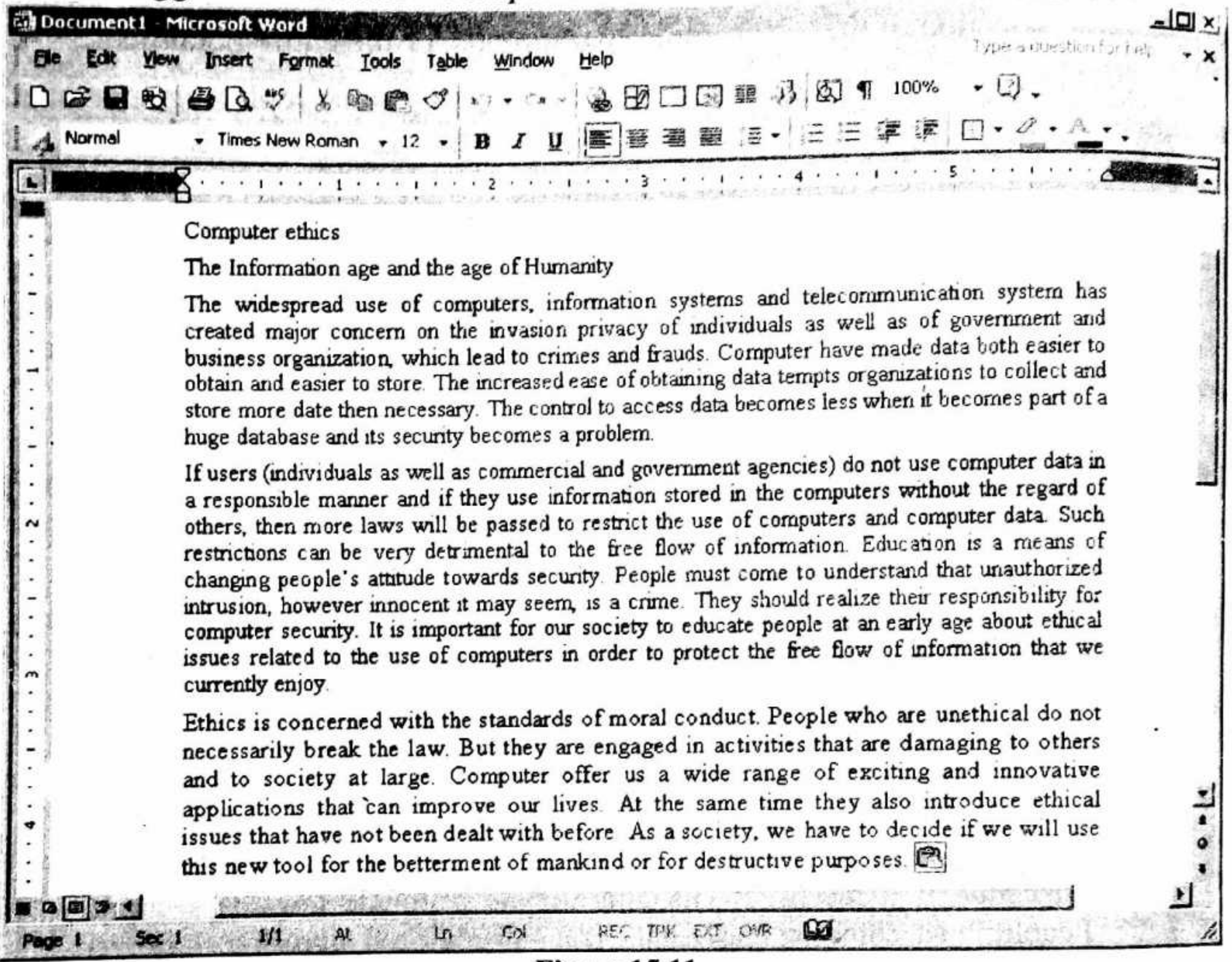


Figure 15.11

3. To select sub-title "The information age and the age of Humanity", move mouse pointer towards left border of the sub-title until it changes to right mouse pointer. CLICK: left mouse button to select the entire line.

The Information age and the age of Humanity

4. To remove this phrase PRESS: DELETE.
5. Move the I-beam pointer to the word " ethics " on the title. Select it by double clicking the left mouse button.
6. Type the word "Security". The word "ethics" will be deleted automatically. The word "Security" will appear.
7. To undo this last editing change, CLICK: Undo button in the standard toolbar. The word ethics reappears.

Let us capitalize the word "Computer ethics".

1. Selected word "Computer ethics".
2. CHOOSE: Change case in the Format Menu drop-down list of options. SELECT: Upper case and PRESS: <ENTER>↵ key or CLICK OK The title "Computer ethics" will change into "COMPUTER ETHICS". (You can also use shift and F3 keys to toggle the selected text between uppercase, lowercase, and initial caps).
3. CLICK: Close button [☒] to close window without saving the changes.

15.7 TEXT FORMATTING

Changing in the appearance of text in a document is referred as text formatting. There are four different levels of text formatting.

CHARACTER FORMATTING

Selection of typeface, font sizes and styles for text is called character formatting. The most common way to emphasize text is to apply **bold** face, *italic* or underline character-formatting styles.

PARAGRAPH FORMATTING

A paragraph varies in length from a single line to a full page. Paragraph formatting includes text alignment; line spacing, indenting, tab settings, numbering and borders etc.

SECTION FORMATTING

Section formatting specifies page numbers, headers and footers for different sections or chapters of a document.

DOCUMENT FORMATTING

Document formatting specifies the overall page layout for printing. It includes choosing the paper size page orientation and margins.

15.8 CHARACTER FORMATTING

Improving the appearance of text is called as character formatting. It involves selecting typeface, font sizes and attributes for text. Character formatting commands can be accessed either through the font dialog box shown in Figure 15.12 on the formatting toolbar, or by using shortcut keyboard

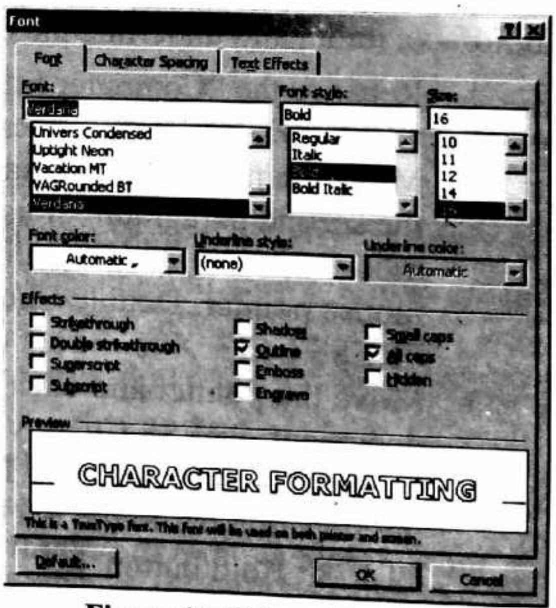


Figure 15.12 Font dialog Box

Scanned by CamScanner

combination, since many of the features are accessible from the formatting toolbar and shortcut keys, you may use option Font on Format command in Menu bar to see a preview of the desired text in the dialog box. Table 15.4 summarizes the mouse and keyboard methods for choosing character formatting commands.

Table 15.4 Character formatting summary

Tool bar button	Key Board Shortcut	Description
B	Ctrl + b	Bolds the selection
I	Ctrl + i	Italicizes the selection
U	Ctrl + u	Underline the selection
	Ctrl + Shift + d	Double under line
	Ctrl + Shift + w	Underline only the word
		Highlighting the selection of:-
Times New Roman	Ctrl + Shift + f	drop-down list of font or type face.
10	Ctrl + Shift + p	drop-down list of font size.
	Ctrl + Shift + a	Capitalizes the selection.
	Ctrl + Shift + k	Make small caps.
	Ctrl + F3	Changes the case of the selection.
	Ctrl + =	Makes a subscript style.
	Ctrl + Shift + =	Makes a superscript style.
	Ctrl + space bar	Remove all character formatting.

Editing and Formatting the document "Computer Ethics"

You can edit and format the document in Word Processing software as you type or after you have selected text. Perform the following steps to apply boldface, italic, and underline the document saved as "computer ethics".

1. Retrieve the "Computer Ethics" document from your drive.
   CHOOSE: Computer ethics from the File pull-down menu list.
2. Move the mouse pointer and click the left mouse button once any where in the second paragraph "If users… enjoy" for the cursor to appear. Move the mouse pointer into selection bar on the left until it changes into right mouse pointer. Double Click the left mouse button, paragraph is selected.
   PRESS Ctrl + X to the cut the selected paragraph.
3. Move the pointer and click the left mouse button once at the bottom of the document. PRESS Ctrl + V to paste the removed paragraph at the bottom.
4. To bold the title, select it. To select the title, position the mouse pointer in the selection bar to the left of the line and click the left mouse button once.
   CLICK: Bold button **B**
   To see the change you have made, CLICK anywhere in the text.

5. To make the sub-title bold, italic and to underline it, select the sub-title.
   CLICK: Bold button
   CLICK: Italic *I*
   CLICK: Underlie button U

Perform the following formatting changes in the first paragraph using keyboard shortcuts.

TEXT TO BE FORMATTED	FORMATTING TO APPLY
Computers, information …Telecommunication system	**bold**
privacy	***italic and bold***
crimes and frauds	*italic*
when it becomes part of……. becomes a problem.	underline

15.9 TYPEFACES, FONTS, AND POINT SIZES

Font refers to the physical characteristics of a typed character. These characteristics include its typeface, point size, style, pitch and spacing.

Typeface: Refers to the design of the characters such as PAKISTAN is an Islamic Country. There are two general categories of fonts:

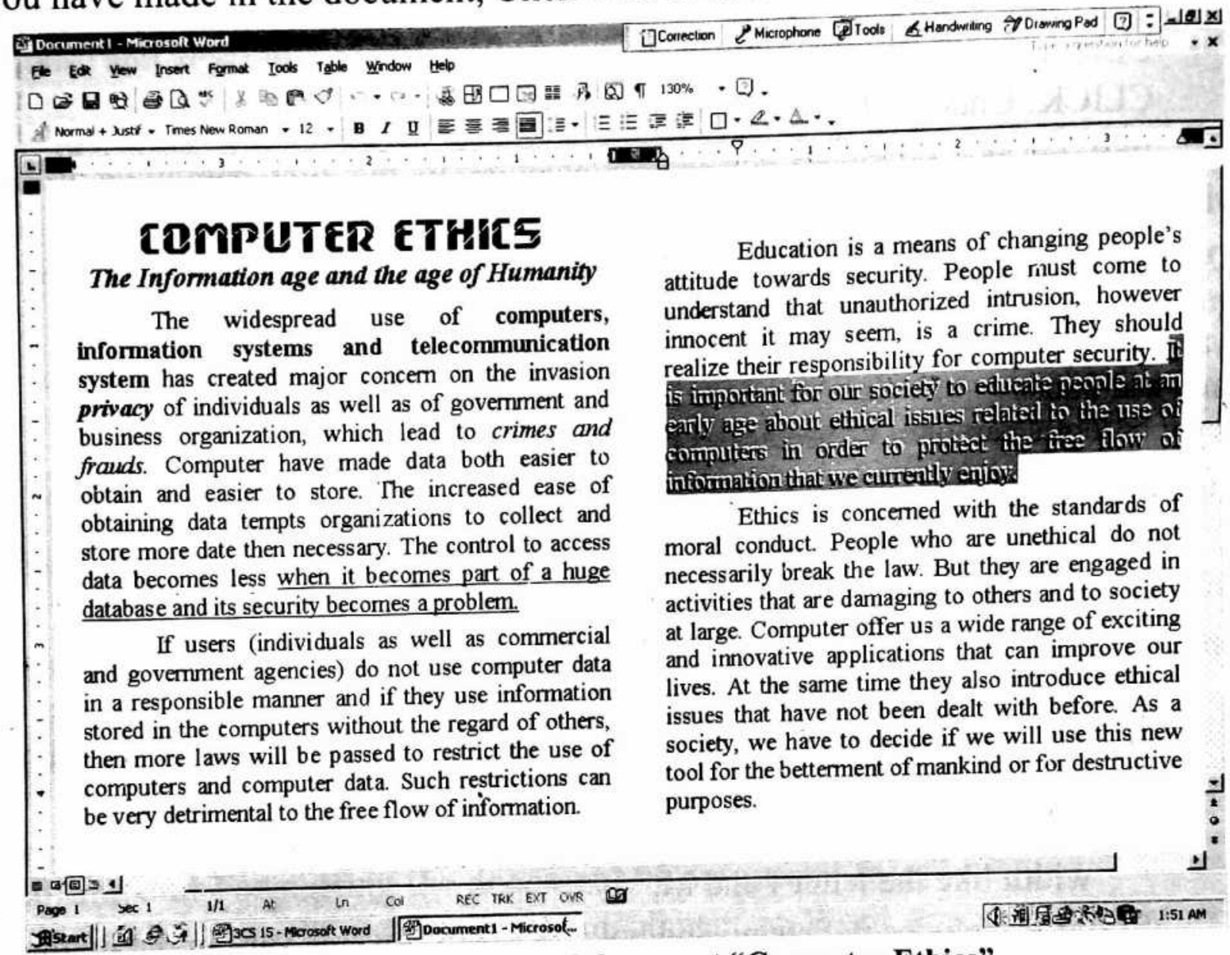
- Mono space fonts in which each character takes up exactly the same space.
- Proportional fonts in which each character has a slightly different width like the letter I and M.

Style: It includes italicising and bold facing. Softwares vary in the number of fonts that can be used. Windows provides easy access to several popular typefaces.

Perform the following steps to change some of the fonts and point size in the computer ethics document.

1. SELECT: The main title "Computer ethics"
2. CLICK: Down arrow beside the font drop down list Times New Roman. Scroll through the font choices by clicking up and down arrow on the scroll bar of drop down list. And SELECT: DexGothicD
3. CLICK: Down arrow beside the font size drop-down list 12.
4. SELECT:22 point font size.
5. SELECT: Sub-title. (Second line the document).
6. CLICK: Down arrow beside the font size and SELECT: 14 on it.
7. PRESS: (HOME) or click any where in the text to remove highlighting.
8. CLICK: Columns icon on the Standard toolbar and highlight two columns. Your document should look as in figure 15.13

When you close the document, a dialog box appears asking you to save changes you have made in the document, Click YES to save the changes in the document.



COMPUTER ETHICS

The Information age and the age of Humanity

The widespread use of **computers, information systems and telecommunication system** has created major concern on the invasion ***privacy*** of individuals as well as of government and business organization, which lead to ***crimes and frauds***. Computer have made data both easier to obtain and easier to store. The increased ease of obtaining data tempts organizations to collect and store more date then necessary. The control to access data becomes less when it becomes part of a huge database and its security becomes a problem.

If users (individuals as well as commercial and government agencies) do not use computer data in a responsible manner and if they use information stored in the computers without the regard of others, then more laws will be passed to restrict the use of computers and computer data. Such restrictions can be very detrimental to the free flow of information.

Education is a means of changing people's attitude towards security. People must come to understand that unauthorized intrusion, however innocent it may seem, is a crime. They should realize their responsibility for computer security. It is important for our society to educate people at an early age about ethical issues related to the use of computers in order to protect the free flow of information that we currently enjoy.

Ethics is concerned with the standards of moral conduct. People who are unethical do not necessarily break the law. But they are engaged in activities that are damaging to others and to society at large. Computer offer us a wide range of exciting and innovative applications that can improve our lives. At the same time they also introduce ethical issues that have not been dealt with before. As a society, we have to decide if we will use this new tool for the betterment of mankind or for destructive purposes.

Figure. 15.13 Formatted document "Computer Ethics"

PARAGRAPH FORMATTING

Paragraph formatting involves changing indentation, alignment, line spacing and tab settings for a paragraph. All these features allow you to enhance the appearance of your document. Paragraph formatting commands are accessible using mouse or keyboard shortcut combinations. For entering specific measurements and accessing the features of paragraph formatting option, select the paragraph command from the format option on the menu bar. The paragraph dialogs box is displayed.

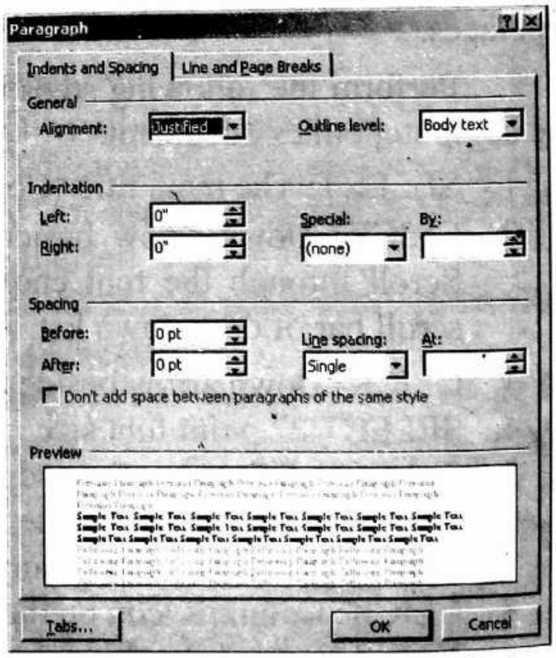


Figure 15.14 Paragraph formatting dialog box

PARAGRAPH FORMATTING SYMBOLS

MS Word stores formatting information in the Paragraph Symbol ¶. To apply paragraph formatting commands, position the insertion point any where in the paragraph and then issue the desire command. To display the paragraph marks and all other symbols, click the show/hide button ¶ on the formatting toolbar. In the paragraph, word shows spaces as dots, tabs as right arrows → and paragraph mark showing the location where the enter key is depressed.

To remove the paragraph marks click the show/hide button ¶ a second time on the formatting toolbar. However, there are nonprinting symbols and will not appear on the printed page.

FORMATTING CHARACTERISTICS

Word has the ability to display help bubble showing the formatting characteristics of the text. To display the help bubble click it on the toolbar or on the Help drop-down list. Position the help button on the desired text area. A side-window will appear on the screen revealing the formatting information.

5.10 INDENTING PARAGRAPH

Indenting a paragraph means shifting the body of text within the normal page margin. When you indent a paragraph, you temporarily change the position of text relative to the left and right margins. You can indent a paragraph on the left side only, on the right side only or on both sides.

Indentation is determine by the tab setting on the Ruler if you use increase indent or decrease indent . Each time you indent a paragraph, the text moves to the next tab stop. If the tab positions have not been modified, word assumes the default 0.5-inch settings. To create larger indentations, you can set larger gaps between tabs or select the indent, command multiple times.

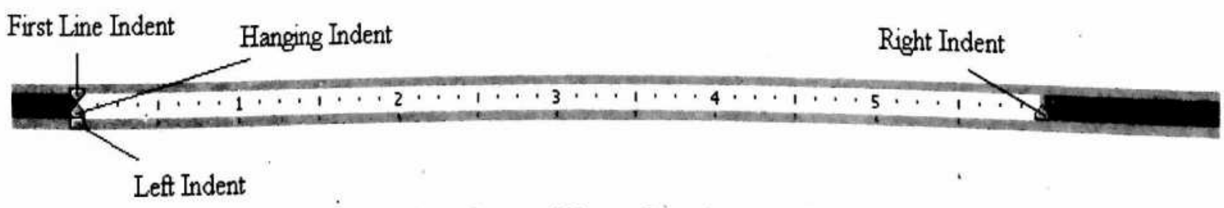


Figure 15.15 **showing the position of various indents on the ruler**

FIRST LINE INDENT MARKER

This indent marker sets the first line of a paragraph on the left margin. It avoids the use of tab key at the beginning of every new paragraph.

LEFT INDENT MARKER

The left indent marker moves the entire paragraph from the left margin.

RIGHT INDENT MARKER

The right indent marker moves the entire paragraph from the right margin.

CHANGING PARAGRAPH ALIGNMENT

Aligning text to the left, right or at the centre of the page affects the outlook of your document. The way how the text is aligned with the margins of a document is called justification. Word provides four different types of justification or alignment.

Toolbar	Description
Left justification	Aligns text at the left margin.
Centre justification	Centres the line or paragraph between the margins.
Right justification	Aligns text at the right margin.
Full justification	spreads the text entry between the left and right margins by automatically spacing the words in a line.

CHANGING LINE SPACING

An important paragraph formatting feature is line spacing that allows you to change the spacing between the lines of a paragraph or the document. You can specify various line spacing options from the paragraph dialog box using the format on the Menu Bar. However the keyboard short cuts listed below provide the quickest way of selecting line spacing in a paragraph.

Short Cut	Description
CTRL + 1	Single space in the selected paragraph
CTRL + 2	Double space in the selected paragraph
CTRL + 5	1.5 line in the selected paragraph

15.11 PAGE/PRINT LAYOUT

MS Word provides three/four primary views for working with documents; Normal, Web layout, Print layout and Outline. You can select your desired view for your document from View pull-down menu on the Menu bar or by clicking the desired View button on left of the horizontal scroll bar shown in Figure 15.18. Your selection for view depends upon the type of work that you perform. Page/print layout view displays a document in almost full preview mode (WYSIWYG i.e., *What You See Is What You Get*). Character, paragraph, and document formatting options are

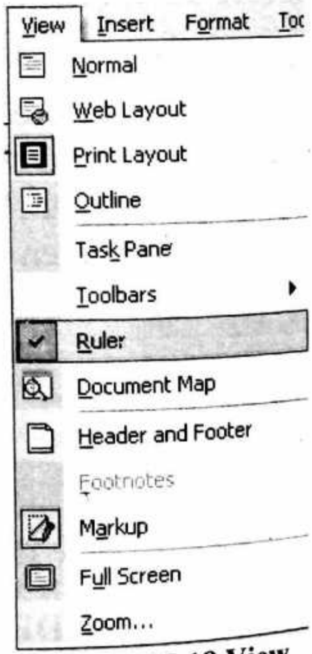


Figure. 15.18 View pull-down menu

all displayed along with headers, footers, and column. The document displayed on screen is what it appear to be on real pages. Table 15.6 gives the descriptions of various views on View drop down menu list.

Table 15.6 Describing various views on View menu list

Menu Command	View button	Description
Normal		Allows you to type, edit and format documents, but does not display additional information such as header and footers.
Web layout		Displays the document as it would appear if published on the web.
Page/Print layout		Allows you to display multiple column, footnotes, header and footers in the document. You can also view graphics in this format.
Outline		Allows you to view and edit the documents structure.

There are two more options for View. These are full screen and zoom.

Full Screen: In Full Screen mode, you can type, edit and format a document without Menu bar, tool bars and Ruler. To return to the regular word workspace from full screen mode, click the Full Screen button once or press [Esc].

Zoom: It is used enlarge or reduce the view of a page on – screen. To modify Word's zoom settings you choose the view, and then zoom command on the drop-down menu list or select the zoom control box 100% on the standard toolbar.

Specifying Page Layout

Page layout of a document is affected by many factors including margins, paper size and print orientation. The control of these factors through page Setup dialog box is accessed from file menu. The page setup dialog box provides four tabs: Margins, Paper size, Paper source and Layout. Figure 15.19 shows the Page Set-up dialog box.

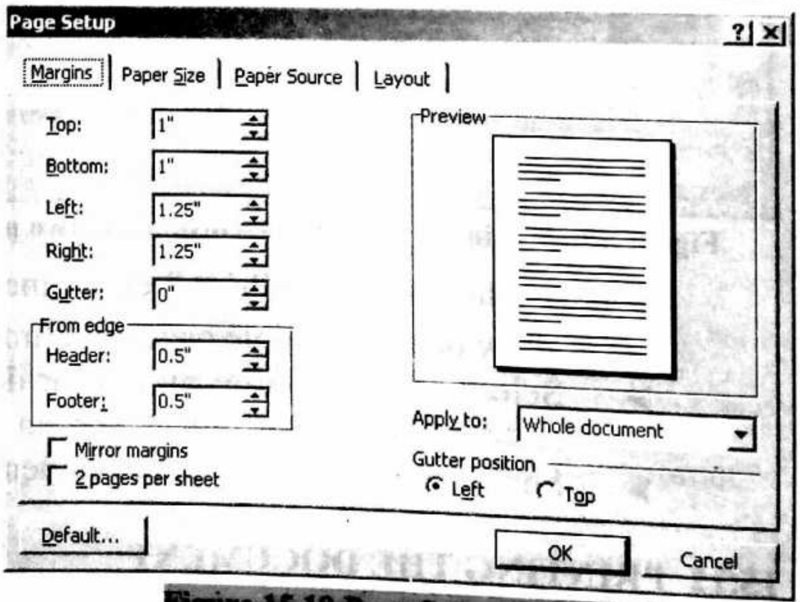


Figure 15.19 Page Setup dialog Box

PREVIEWING THE DOCUMENT

Before you pass a document for printing, it is better to look at the document how it would appear when printed. Word provides the facility to preview document in full-page display before sending them to the printer. To access word's Print Preview mode, you can select Print Preview option from the File Menu on the Menu bar or click the Print Preview button on the standard toolbar. The Print Preview screen appears as shown in Figure 15.20.

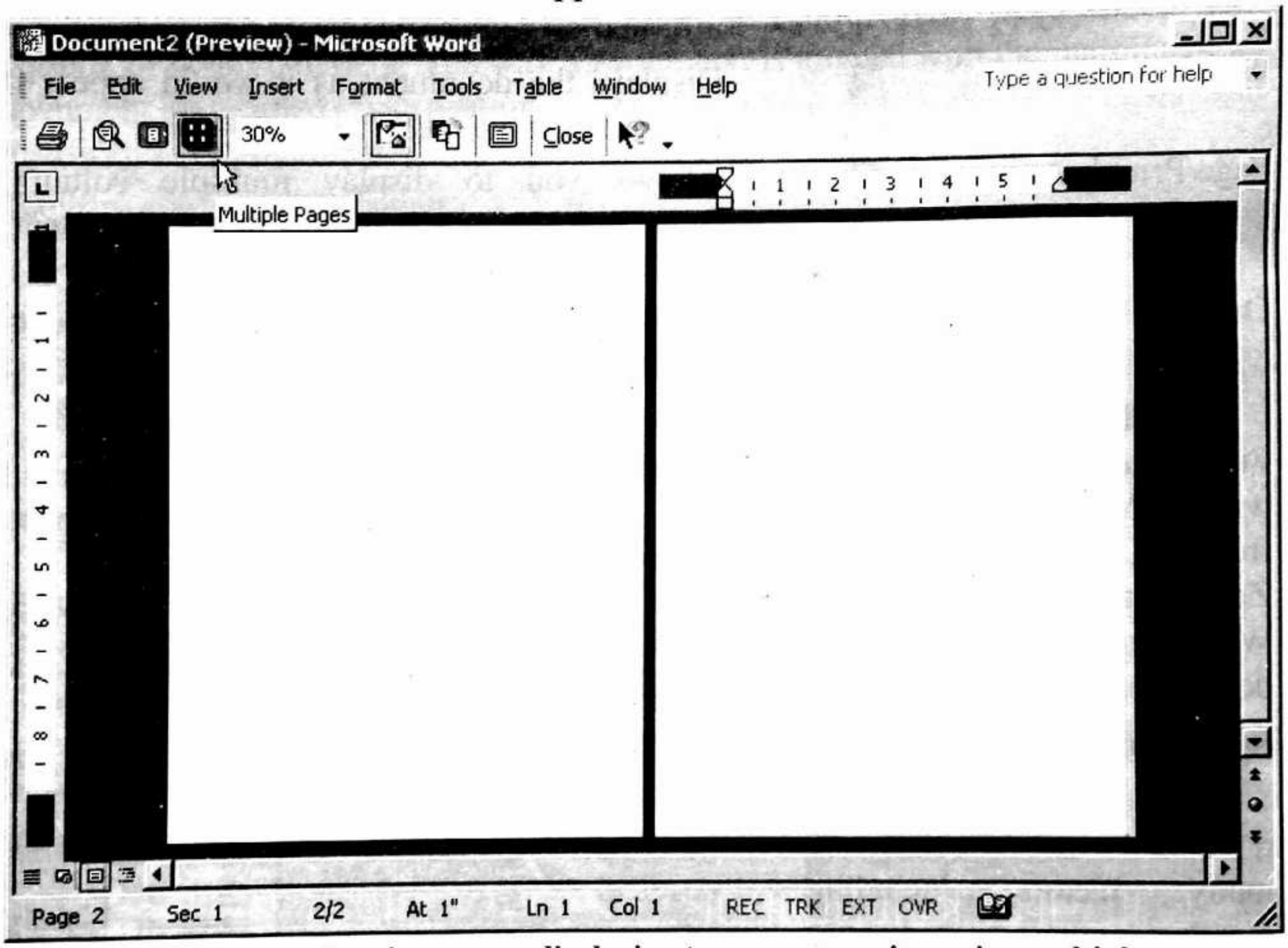


Figure 15.20 Print Preview screen displaying two pages preview using multiple pages

Some of the features of Print Preview include the following:-

- View one or multiple pages at a time.
- Select Zoom mode to view your document or Edit mode to modify and format your document without leaving Print Preview.
- Shrink to Fit mode to fit a document into a single page.

15.12 PRINTING THE DOCUMENT

Open the document you want to print. If you are satisfied with your document after viewing it in page layout, and Print Preview, then you can send the document to the printer attached with your computer. The quickest and the

fastest method for printing a single copy of every page in a document is to click the print button on standard toolbar. If you want to print specific pages only or if you want multiple copies of a document, you must choose the print command from the drop-down list of File Menu and make your selections in the print dialog box shown in Figure 15.21.

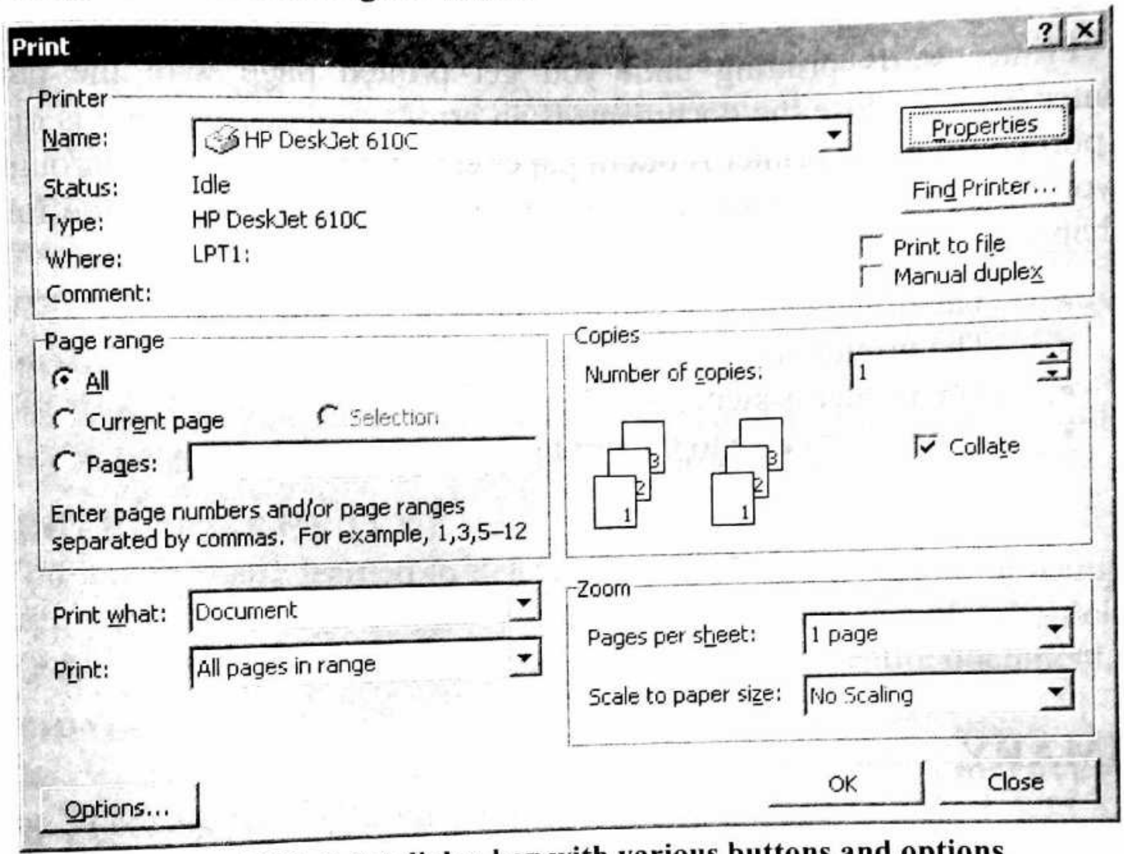


Figure 15.21 Print dialog box with various buttons and options

Make the appropriate setting such as:-

PAGE RANGE AREA

- To print the entire document CLICK: All.
- To print the current page (the page on which cursor is positioned) CLICK: current.
- To print selected pages. CLICK: pages (Enter the numbers of pages selected)

COPIES AREA

- Enter the number of copies of each page(s) required using the spinner button.
- You can collate or deselect the collate option, the way in which pages are printed out.

Perform the following steps to print a document:

1. To print the "Computer Ethics" document.
   SELECT: print command from File menu list.
2. Print dialog box appear with option All Pages in range and number of pages as 1 in copies area.
   CLICK: OK

Printer starts printing until you get printed page with the document Computer Ethics. Close the document. If an error occurs, a message is displayed like a port conflict, the printer is out of paper etc. If you are printing through serial port, you will see a printer out of Paper message when any one of the following thing happens.

- The printer is out of paper.
- The printer is offline
- The printer is switched off.
- The power cable to the printer is disconnected.

SUMMARY

Word processing refers to the use of a computer program to prepare and print documents.

MICROSOFT WORD

MS Words provides facilities to prepare, modify and print documents. Some of these are book marks, autotext, autotable, links, pictures, symbols, office clipboard, video clips, header and footer, footnotes, fonts, borders, shading, columns, spell checker, grammar checker, autocorrect, thesaurus etc.

COMPONENTS OF MICROSOFT WORD

- The top most band is called **the title bar**.
- Second band is called **the Menu bar**.
- Third band is called **the standard toolbar**.
- Fourth band is called **the formatting toolbar**.
- Below the toolbars appears **the ruler bar**
- Below and on the right border of the document are **the scroll bars**.
- **Status bar** is located at the bottom of the application window

EDITING A DOCUMENT

Editing is the process of making alterations in the text. Some features of editing are insert, delete and undelete, find and replace, cut/copy and paste etc.

SAVING A DOCUMENT

A document can be saved on a disk in a number of ways such as using save button on the standard toolbar or selecting save command from the File drop down menu. By default, while working in Word, it automatically saves a document after every ten minutes.

FORMATTING DOCUMENTS

Document formatting is a process of improving the over all appearance of a document. A formatted document can communicate a message more effectively

SELECTING TEXT AND BLOCKING TEXT

MS Word can be used to select a block of text. This refers to blocking the text. This can be done either by mouse action or through keyboard short cuts.

DELETING BLOCKS OF TEXT

You have already learned to delete text one character at a time using the Back Space and Delete keys. Similarly larger blocks of text can also be deleted after making selection of the words, lines, paragraphs, or the entire document.

TEXT FORMATTING

The changing of the appearance of text in a document is referred to as text formatting. There are four different levels of text formatting.

Character Formatting

Selection of typeface, font sizes and styles for text is called character formatting.

Paragraph Formatting

Paragraph formatting includes text alignment; line spacing, changing indent, tab settings, numbering, positions and fonts of headings and sub-headings etc.

Section Formatting

Section formatting specifies page numbers, headers and footers for different sections or chapters of a document.

Document Formatting

Document formatting specifies the overall page layout for printing. It includes choosing the paper size page orientation and margins.

Typefaces, Fonts, And Point Sizes

Font refers to the physical characteristics of a typed character. These characteristics include its typeface, point size, style, pitch and spacing.

Style: Includes italicising and bold facing. Softwares vary in the number of fonts that can be used.

Paragraph Formatting Marks

MS Word stores formatting information in the Paragraph Symbol ¶. In a paragraph, word shows spaces as dots, tabs as right arrows → and paragraph mark ¶ showing the location where the enter key is depressed.

Indenting Paragraph

Indenting a paragraph means shifting the body of text within the normal page margin on the left side only, on the right side only or on both sides.

Changing Line Spacing

An important paragraph formatting feature is line spacing that allows you to change the spacing between the lines of a paragraph or the document.

PAGE/PRINT LAYOUT

MS Word provides three primary views for working with documents. Normal, Web layout, page/print layout and outline.

Zoom: Allows enlarging or reducing the view of a page on the screen.

Specifying Page Layout

Page layout of a document is affected by many factors including margins, paper size and print orientation. MS Word provides the control of these factors through page Setup dialog box that is accessed through file menu.

PREVIEWING THE DOCUMENT

Word provides the facility to preview document in full-page display before sending them to the printer or to view multiple pages at the same time.

PRINTING THE DOCUMENT

One of the methods for printing a single copy of every page of a file is to click the print button on standard toolbar. If you want to print only the specific pages or if you want multiple copies of a document in a specific order then you must choose the print command from the drop-down list of File Menu.

EXERCISE

15.01 Complete the following statements

i) MS Word is a windows based ___________

ii) Press the _________ key for over type mode.

iii) There are _______ items on the menu bar of MS Word.

iv) Press control + shift + = to make __________style.

v) Press CTRL + 5 to increase_____________in the selected paragraph.

15.02 Tick (✓) the following statements either True or False.

i)	In MS Word any characters can be used in a filename.	True/False
ii)	To select the entire document triple click the document.	True/False
iii)	Ctrl+shift+d is used to drop the capitals in the select text.	True/False
iv)	In paragraph formatting the word shows spaces as dots.	True/False
v)	In case of 10 point font a maximum of 5 lines can be typed in an inch.	True/False

15.03 Encircle one choice A, B, C or D in each case.

i) Second bar on MS Word Window is called:-

(A) Menu Bar (B) Tool Bar
(C) Status Bar (D) Formatting tool bar

ii) Pressing CTRL+V will cause:

(A) To appear View drop-down list. (B)To past the copied item.
(C) Full screen view of the Document. (D) To cut the selected item

iii) Toolbars command is listed on drop-down list of:

(A) View (B) Insert (C) Format (D) Tools

iv) By default, word automatically saves your document after every

(A) 5 minutes (B) 10 minutes (C) 15 minutes (D) 30 minutes

v) Pressing CTRL+SHIFT+A causes:-

(A) To select whole document (B) To highlight the selected Text
(C) To capitalize the selected text (D) Office Assistant to appear

15.04 Match the items given in Column I with those in Column II

	Column I		Column II
i)	Change Case	a)	Formatting Toolbar
ii)	Help	b)	Insert
iii)	Edit	c)	Shift +F3
iv)	Symbol	d)	Standard Toolbar
v)	Left justification	e)	F1
		f)	Office Clipboard

15.05. What are the advantages of using a word processing program over a typewriter?

15.06. List the advantages of working MS Windows environment.

15.07. What are the different levels of formatting a document?

15.08. Why is it significant to know that the default mode in word is the Insert mode?

15.09. Why is it important to close a document before retrieving another file?

15.10. What happen if you press [enter] when the insertion point is in the middle of a paragraph?

15.11. How will you delete a character to the left of the insertion point?

15.12. What is meant by the term Word wrap?

15.13. What is the difference between the File; Save and File, Save As command?

15.14. How do you select an entire document using the mouse?

15.15. What are the keyboard shortcut keys for applying bold, italic and under lines the text?

15.16. What is the difference between typeface and a font?

15.17. What are the different paragraph alignment options?

15.18. Describe some methods for moving the cursor around a document.

15.19. Name three different views you can access by clicking buttons on the horizontal scroll bar.

15.20. How do you return to Word's application window when you are in Full Screen mode?

15.21. Name the tabs in Page set-up dialog box.

15.22. What are the two methods for changing the margins in a document?

15.23. How do you preview a document before sending it to the printer?

15.24. What are the two modes available in Print Preview?

ANSWERS TO EXERCISES

CHAPTER 1

1.01 Complete the following statements.

i) data processing (ii) abacus (iii) Babbage
iv) fourth (v) Internet (vi) Universal Product code.
vii) Assembly (viii) Beginner's All- purpose Symbolic Instructional code.
ix) Machine language (x) symbolic Icons

1.02 Tick (✓) the following statements either True or False.

i) False ii) False iii) True iv) False v) True

1.03 Encircle one choice A, B, C or D in each case.

i) B ii) B iii) D iv) B v) A

1.04 Match the items given in Column I with those given in Column II

i) g ii) h iii) a iv) b v) j
vi) i vii) c viii) e ix) d x) f

CHAPTER 2

2.01 Complete the following statements

i) CPU memory unit. ii) Central processing Unit.
iii) Arithmetic and logic Unit. iv) registered. v) 2

2.02 Tick ( ) the following statements either True or False.

i) False ii) True iii) False iv) True (v) False

2.03 Encircle one choice A, B, C or D in each case.

i) (B) ii) (C) iii) (A) iv) (D) v) (D)

2.04 Match the items given in Column I with those in Column II

i) (c) ii) (d) iii) (e) iv) (b) v) (a)

CHAPTER 3

3.01 Complete the following statements

i) QWERTY. ii) various options. iii) mouse
iv) voice recognition. v) output

3.02 Tick (✓) the following statements either True or False.

i) False ii) False iii) True iv) True v) False

3.03 Encircle one choice A, B, C or D in each case.

i) (C) ii) (A) iii) (B) iv) (C) v) (A)

3.04 Match the items given in Column I with those in Column II

i) (e) ii) (c) iii) (d) iv) (a) v) (b)

CHAPTER 4

4.01 Complete the following statements

i) 8 (ii) RAM (iii) 512 (iv) Hard disk (v) 500

4.02 Tick (✓) the following statements either True or False.

i) True (ii) False (iii) False (iv) True (v) False

4.03 Encircle one choice A, B, C or D in each of the Multiple choice questions.

i) (A) (ii) (B) (iii) (C) (iv) (A) (v) (B)

4.04 Match the items given in Column I with those given in Column II

i) (c) (ii) (a) (iii) (d) (iv) (e) (v) (b)

CHAPTER 5

5.01 Complete the following statements

i) raw facts (ii) arranged (iii) 142 (iv) 0011110 (v) 1
(vi)American Standard Code for Information Interchange

5.02 Tick (✓) the following statements either True or False.

i) False (ii) True (iii) False (iv) True (v) False

5.03 Encircle one choice A, B, C or D in each case.

i) (A) (ii) (C) (iii) (A) (iv) (C) (v) (B)

5.04 Match the items given in Column I with those in Column II

i) (c) (ii) d) (iii) e) (iv) a) (v) (b)

5.07 i) 101111 (ii) 10100111 (iii) 0.111 (iv) 10101.101

5.08 i) 45 (ii) 78 (iii) 0.71875 (iv) 5.3125

5.09	i) 53	(ii) 443	(iii) 250	(iv) 3732	
5.10	i) 23	(ii) 250	(iii) 577	(iv) 1253	
5.11	i) 25	(ii) 8D	(iii) 214	(iv) 2BE	
5.12	i) 79	(ii) 47692	(iii) 65453	(iv) 10726	
5.13	i) 11_2	(ii) 1101_2	(iii) 110_2	(iv) 1011_2	v) 111111_2

CHAPTER 6

6.01 Complete the following statements

i) OR (ii) multiplication (iii) cancellation (iv) 1 (v) (A+B).(A+C)

6.02 Tick (✓) the following statements either True or False.

i) False (ii) False (iii) True (iv) True (v) False

6.03 Encircle one choice A, B, C or D in each case.

i) (C) (ii) (C) (iii) (C) (iv) (C) (v) (C)

6.04 Match the items given in Column I with those given in Column II

i) (e) (ii) (c) (iii) (a) (iv) (b) (v) (d)

6.07 i) 1 (ii) 0 (iii) 1

CHAPTER 7

7.01 Complete the following statements

i) operating system (ii) service (iii) COMMAND.COM (iv) A

v) internal (vi) external

7.02 Tick (✓) the following statements either True or False.

i) False (ii) False (iii) True (iv) True (v) True

vi) True

7.03 Encircle one choice A, B, C or D in each case

i) (A) (ii) (D) (iii) (D) (iv) (B) (v) (B

vi) (D)

7.04 Match the items given in Column I with those given in Column II

i) (d) (ii) (e) (iii) (g) (iv) (a) (v) (b)

CHAPTER 8

8.01 Complete the following statements

i) command line (ii) graphic user's (iii) interact (iv) Start (v) exit

8.02 Tick (✓) the following statements either True or False.

i) False (ii) False (iii) True (iv) True (v) True

8.03 Encircle one choice A, B, C or D in each case.

i) (C) (ii) (C) (iii) (D) (iv) (C) (v) (A)
vi) (C)

8.04 Match the items given in Column I with those given in Column II

i) (d) (ii) (e) (iii) (f) (iv) (j) (v) (a)
vi) (i) (vii) (g) (viii) (d) (ix) (h) (x) (b)

CHAPTER 9

9.01 Complete the following statements

i) direct, program (ii) direct (iii) reserve word
(iv) terminal Symbol (v) control (vi) algorithm(vii) debugging

9.02 Tick (✓) the following statements either True or False.

i) True (ii) False (iii) False (iv) False (v) True
vi) False (vii) True

9.03 Encircle one choice A, B, C or D in each case.

i) (B) (ii) (D) (iii) (C) (iv) (B) (v) (D)

9.04 Match the items given in Column I with those given in Column II

i) (f) (ii) (g) (iii) (a) (iv) (e) (v) (h)
vi) (b) (vii) (d) (viii) (c)

CHAPTER 10

10.01 Complete the following statements

i) Beginner's All-purpose Symbolic Instruction Code (ii) limited
iii) LIST (iv) Statements (v) F2 (vi) Input

10.02 Tick (✓) the following statements either True or False.

i) False (ii) True (iii) True (iv) False (v) True
vi) True (vii) False (viii) False (ix) True (x) True

10.03 Encircle one choice A, B, C or D in each case.

i) (D) (ii) (C) (iii) (D) (iv) (C) (v) (D) (vi) (C)

10.04 Match the items given in Column I with those given in Column II

i) (d) (ii) (e) (iii) (g) (iv) (h) (v) (b) (vi) (a)

CHAPTER 11

11.01 Complete the following statements

i) Control (ii) Out of data in iii) many times
iv) Controlled (v) FOR … NEXT

11.02 Tick (✓) the following statements either True or False.

i) True (ii) False (iii) False (iv) True (v) True

11.03 Encircle one choice A, B, C or D in each case

i) (A) (ii) (A) (iii) (B) (iv) (C) (v) (B) (vi) (D)

11.04 Match the items given in Column I with those given in Column II

i) (d) (ii) (f) (iii) (e) (iv) (g) (v) (b) (vi) (a)

CHAPTER 12

12.01 Complete the following statements

i) one dimensional (ii) filling (iii) 11 (iv) 50 (v) dimension

12.02 Tick (✓) the following statements either True or False.

i) True (ii) False (iii) True (iv) False (v) False

12.03 Encircle one choice A, B, C or D in each case.

i) (D) (ii) (D) (iii) (B) (iv) (C) (v) (B)

12.04 Match the items given in Column I with those given in Column II

i) (f) (ii) (e) (iii) (g) (iv) (a) (v) (c) (vi) (b)

CHAPTER 13

13.01 Complete the following statements

i) once (ii) save (iii) module (iv) RETURN (v) OPEN

13.02 Tick (✓) the following statements either True or False.

i) True (ii) False (iii) True (iv) False (v) True

13.03 Encircle one choice A, B, C or D in each case.

i) (C) (ii) (D) (iii) (D) (iv) (D) (v) (B)

13.04 Match the items given in Column I with those given in Column II

i) (c) (ii) (d) (iii) (e) (iv) (a) (v) (b)

CHAPTER 14

14.01 Complete the following statements

i) pixels (ii) 640×200 (iii) previous coordinate (iv) move upward (v) N (vi) Input

14.02 Tick (✓) the following statements either True or False.

i) False (ii) True (iii) True (iv) False (v) True

14.03 Encircle one choice A, B, C or D in each case.

i) (C) (ii) (A) (iii) (B) (iv) (C) (v) (D)

14.05 Match the items given in Column I with those given in Column II

i) (d) (ii) (f) (iii) (e) (iv) (b) (v) (a) (vi) (a)

CHAPTER 15

15.01 Complete the following statements

i) GUI (ii) Insert (iii) 9 (iv) superscript (v) 1.5 line

15.02 Tick (✓) the following statements either True or False.

i) False (ii) True (iii) False (iv) True (v) False

15.03 Encircle one choice A, B, C or D in each case.

i) (D) (ii) (C) (iii) (D) (iv) (C) (v) (D) (vi) (C)

15.04 Match the items given in Column I with those given in Column II

i) (c) (ii) (e) (iii) (f) (iv) (b) (v) (a)

GLOSSORY

A

Abacus An ancient calculating device consisting of moveable beadson wires fixed in a frame.

,**ABC** The first electronic digital computer built by Dr. John Atanasoff and Clifford Berry in 1942.

Access Time The time required to retrieve information from the computer.

ADA A new programming language developed in late 1970s.

Address A number specifying where a unit of information is stored in the computer's storage.

Algorithm A set of rules for solving a problem.

Allocation The process of reserving computer storage areas for instructions or data.

Analog computer A device that operates on data in the form of continuously variable physical quantities.

Antivirus utility A program that scans disks and memory for viruses, detects, and remove them.

Application program A computer program designed to solve a particular type of problem or perform a specific operation, such as inventory control, or word processing.

Arithmetic-logic unit The portion of the central processing unit where arithmetic and logical operations are performed.

Array (1) A series of related items. (2) An ordered arrangement or pattern of items or numbers, such as a determinant, matrix, vector, or a table of numbers.

Artificial Intelligence A branch of computer science that involves computers to simulate the thinking of human beings.

ASCII code American Standard Code for Information Interchange. The standard code used for information interchange among data processing systems, communication systems, and associated equipment.

Assembler A computer program that takes non-machine language instructions prepared by a computer user and converts them into a form that may be used by the computer.

Assembly language A programming language which allows a computer user to write a program using mnemonics instead of machine code. It is a low-level symbolic programming language which closely resembles machine code language. The language uses groups of letters; each group represents a single instruction.

Automation Activity that is accomplished with little or no human intervention.

B

Back-up A copy of a program or document used in case of any damage/failure of the original one.

Base The radix of a number system.

BASIC Beginner's All-purpose Symbolic Instruction Code. An easy-to-learn,

easy-to-use high level programming language.

Baud A unit for measuring data transmission speed.

Binary The basis for calculations in all computers, this two-digit numbering system consists of the digits 0 and 1.

Binary device (1) A device that can register two conditions; e.g., an electrical switch which can be ON or OFF. (2) In computer science, equipment that records data in binary form or that reads the data so coded.

Binary system A numeral system with a base or radix two; e.g., the numeral 111 represents 1×2^0, plus 1×2^1, plus 1×2^2 (i.e.,7 in base ten).

Bit The smallest unit of information that the computer recognizes, a bit (binary digit 0 or 1) by the presence or absence of an electronic pulse.

Bitmap A binary representation of an image in which each part of the image, such as pixel, is represented by one or more bits in a coordinate system.

Boolean algebra A branch of symbolic logic which is similar in form to algebra but, instead of numerical relationships, it deals with logical relationships.

Boot The process of starting the computer.

Bubble memory A method by which data is stored as magnetized dots (bubbles) which rest on a thin film of semi-conductor material. Offers a compact storage capability.

Buffer A temporary storage area which is used to equalize or balance the different operating speeds. For example, a buffer can be used between a slow input device, such as a keyboard, and the main computer which operates at a very high speed.

Bug A term used to denote a mistake in a computer program or system, or a malfunction in a computer hardware component. Debugging is the removing mistakes and correcting malfunctions.

Bus A channel or path (of wires) for transferring data and electrical signals.

Buttons Symbols found in graphical environments that simulate a push button; the user clicks a button to initiate an action.

Byte A grouping of eight adjacent binary digits operated on by the computer as a unit.

C

C The full name of a programming language designed for use on microcomputers. The language combines high-level statements with low-level machine control to deliver software that is both easy to use and highly efficient.

CAD Computer-Aided Design,

CAI Computer-Assisted Instruction in which computer is used as an instructional tool.

CAM Computer Aided Manufacturing.

Cathode ray tube An electronic tube with a screen upon which information may be displayed. Abbreviated CRT.

Central processing unit(CPU) The brain of a computer system that controls all the components of a computer and process data.

Character Any symbol, digit, letter, or punctuation mark stored or processed by computing equipment.

Chip A thin silicon wafer on which electronic components are fabricated to form an integrated circuit.

Clipboard A holding area maintained by the operating system in memory; used for storing text, graphics, sound, or video that has been copied or cut from a document.

Clock A timing device that generates the basic periodic signal used to control the timing of all operations in a computer.

COBOL Common Business Oriented Language, a higher-level language developed for business data processing applications.

Code A set of rules outlining the way in which data may be represented.

Coding Writing instructions which will cause a computer to perform some specified operations.

Comments Verbal explanations added to a program for purposes of documentation.

Compare To examine two quantities, usually for the purposes of determining identity or relative magnitude.

Compatible A quality possessed by a computer system which enables it to handle both data and programs devised for some other type of computer system.

Compile To prepare a machine language program (or a program expressed in symbolic coding) from a program written in another higher-level programming language, such as FORTRAN or BASIC.

Compiler A computer program that translates a source program written in a higher-level language by a computer user into a machine language program.

Computer A general-purpose machine that process data according to the instructions and produces the result as output.

Computer-aided design Use of the computer for industrial design and technical drawing.

Computer art Art form produced by computing equipment.

Computer-assisted instruction (CAI) The use of the computer to aid in the instruction process. CAI is responsive to the individual needs of the individual student.

Computer-based learning (CBL) A term used to embrace all the present forms of educational computing.

Computer graphics Converting digital information into a format that can be displayed visually on a graphics terminal.

Computer literacy A basic understanding of computer systems and concepts.

Computer security Involves the protection of computer system equipment and data from unauthorized access.

Computer simulation Representing a system or a process by a computer model constructed from a computer program.

Computer system The physical equipment and instructions, i.e., hardware and software, used as a unit to process data.

Computer word A fixed sequence of bits, bytes, or characters treated as a unit and capable of being stored in one storage location.

Computerized data base A set of computerized files on which an organization's activities are based.

Constant A value that does not change during the execution of the program.

Counter A register or computer storage location used to represent the number of occurrences of an event.

Crash The failure of software or hardware leading to abnormal cessation of processing.

Cursor A position indicator on a display terminal to indicate the position of entering data

Cybernetics the use of computer technology to create virtual space.

D

Data A representation of facts or concepts, in a formalized manner suitable for communication, interpretation, or processing by humans or automatic means.

Data base A collection of libraries of data.

Data processing One or more operations performed on data to achieve a desired objective.

Data storage devices Units for storing large quantities (millions) of characters. Typically magnetic disk units, magnetic tape units, magnetic drums, and mass storage devices.

Debug To detect and eliminate all mistakes in a computer program and any malfunctions in the computing system itself.

Decimal number A numeral, usually of more than one digit, representing a sum, in which the quantity represented by each digit is based on the radix of ten.

Decision symbol A flowcharting symbol used to indicate a choice or branching in the information processing path. A diamond shaped figure is used to represent this symbol.

Density The number of characters that can be stored in a given physical space, e.g., an inch of magnetic tape or a square inch of a disk.

Desktop The graphical representation of a person's workplace in which all files, utilities, and programs are at his/her fingertip.

Digital computer A device that manipulates digital data and performs arithmetic and logic operations of these data.

Disk A revolving plate upon which data and programs are stored.

Diskette A floppy disk. A low cost bulk storage medium for micro- and mini-computers.

Display A visual representation of data.

Display unit A device which provides a visual representation of data.

DOS Disk Operation System. An operating system that uses disks to assemble, edit, and execute programs.

E

EBCDIC See Extended Binary Coded Decimal Interchange Code.

EDP Electronic Data Processing

ENIAC. An early electronic computer called Electronic Numerical Integrator and Calculator, built in 1946 by John Mauchly and J. Prosper Eckert at the University of Pennsylvania.

Extended Binary Coded Decimal Interchange Code (EBCDIC) A coding method used to represent

numbers, letters, special characters, and other symbols.

F

File A collection of related records treated as a unit.

File allocation table (FAT) A table log created in the logical formatting process that records the location of each file and the status of each sector.

Firmware A program, contained on a silicon chip, that combines elements of hardware and software.

First generation The first commercially available computers, introduced with UNIVAC I in 1951, and terminated with the development of the transistor in 1959. First generation computers are characterized by their use of vacuum tubes.

Floating point A form of number representation in which quantities are represented by a number called the mantissa multiplied by a power of the number base.

Flowchart A diagram that uses symbols and interconnecting lines to show the logical flow of insrtructions.

Flowchart symbol A symbol used to represent operations, data, flow, or equipment on a flowchart.

Flowchart template A plastic guide containing cutouts of the flowchart symbols that is used in the preparation of a flowchart.

Format The specific arrangement of data.

FORTRAN Formula translator. A higher-level programming language used to perform mathematical, scientific, and engineering computations. A widely used programming language.

Fourth generation computer A modern digital computer that uses large scale integration (LSI) and very large scale integration (VLSI) circuitry.

G

Graphics A feature that permits the construction of lines and other geometric shapes on a display or plotting device.

GUI Graphical user interface in which actions are initiated when the user selects an icon or an option from a menu list with a pointing device.

H

Hacker An expert in computer technology who uses skill and innovative techniques to solve complex computing.

Hard copy A printed copy of machine output in readable form, for example, reports, listings, or documents.

Hardware The physical components of the computer system; for example, mechanical, electrical, or electronic devices.

Hexadecimal number A numeral, usually of more than one digit, representing a sum in which the quantity represented by each digit is based on a radix of sixteen. The digits used are 0, 1,2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D. E, and F,

Holographic storage Uses the laser beam to create images for computer storage.

Host computer The controlling computer in a network.

Hybrid computer A special-purpose computer capable of both analog and digital processing and used mainly for simulation applications.

IC Integrated Circuit, An electronic circuit or combination of circuits contained on semiconductor material; the basis of a computer's intelligence.

Indent The distance between the beginning or end of a line of text and the margin.

Information Meaning assigned to data by humans.

Information science The knowledge of creating, communicating and using, information in all forms.

Input The data that is entered into the computer; the act of entering data.

Input device A unit used to get data/instruction from the user, and transfer it to the computer.

Instruction A group of characters, bytes, or bits that directs a computer to perform an operation.

Interface A common boundary between two pieces of hardware (e.g., a disk unit and a microcomputer) or between two computer systems.

Intrepreter A computer program that translates each statement of source program into a sequence of machine instructions and executes these machine instructions one by one before translating the next source language statement.

Iterate To repeat, automatically, under program control. The same series of processing steps until a predetermined stop or branch condition is reached.

J

Joystick A device for inputting X,Y coordinates by movements of a lever to drive the motions of a cursor on a graphical screen—up, down, left, and right,

Justify Align the characters in a field. For example, left justify makes the first character (e.g., the most significant digit) appears in the left-most character position in a field. right justify makes the last character (e.g., the least significant digit) appears in the last or right-most character position in the field.

K

K Computer shorthand for the quantity 1024; the term is generally used as a measurement of computer memory capacity.

Kerning A text editing feature that adjusts the distance between individual letters in a word to make the word easier to read.

Keyword A primary element in a programming language statement; e.g., words such as LET, GOTO and INPUT in the BASIC programming language.

Kilobyte A kilobyte is 2^{10} or 1024 bytes. It is commonly abbreviated to "K" and used as a suffix when describing memory size. Thus 24K really means a 24 x 1024 = 24576 byte memory system.

L

Language A set of rules, representations, and conventions used to convey information.

Large scale integration (LSI) The process of placing a large number of integrated circuits on one silicon chip.

Least significant digit Pertaining to the digit of a number that has the least weight or significance; e.g., in the number 54321, the least significant digit is 1. Abbreviated LSD.

Load To read information into the storage of a computer.

Location Loosely, anyplace in which data may be stored.

Log on To access a computer system.

Loop A sequence of instructions in a program that can be executed repetitively until certain specified conditions are satisfied.

Low-level language Any programming language that approximates machine language more closely than it does human language.

LSI Large Scale Integration. The process of integrating a large number of circuits on a single chip of semiconductor material.

M

Machine language The basic language of a computer. Programs written in machine language require no further interpretation by a computer.

Magnetic Ink Character Recognition The recognition of characters printed with a special magnetic ink by machines. Abbreviated MICR.

Main storage The fastest general purpose storage of a computer.

Memory The section of the computer where instructions and data are stored; synonymous with storage.

Menu A set of options listed on a terminal display—the user may select from the list those he/she desires.

Merge To combine items into one sequenced file from two or more similarly sequenced files without changing the order of the items.

Microcomputer A small, low cost computer in which the central processing unit (CPU) is an integrated circuit deposited on a silicon chip.

Microprocessor An integrated circuit that will perform a variety of operations in accordance with a set of instructions. Microprocessors are used widely as the control devices for business machines, game machines, household appliances, automobile systems, and microcomputers.

MIS Management Information System. A system designed to provide information to management as an aid to decision making.

Modem An acronym for modulator-demodulator; a device used at each end of a telephone line to convert binary digital data to audio tones suitble for transmission over the line and vice versa.

Motherboard The main circuit board of the computer that contains the CPU, memory, expansion slots etc linked with bus. Also called system board.

N

Network A system of interconnected computer systems and/or terminals. Two large microcomputer consumer networks are compuserve Information Service and The Source.

O

OCR Optical Character Recognition. Characters printed in a special type style which can be read by both machines and people.

Octal Pertaining to a number system with a radix of eight, Octal numbers are frequently used to represent binary numerals, with each octal digit representing a group of three binary digits (bits); e.g., the binary numeral 111000010001101 can be represented as octal 70215.

Off-line A term describing equipment, devices, or persons not in direct communication with the central processing unit of a computer. Equipment which are not connected to the computer.

On-line A term describing equipment, devices, and persons that are in direct communication with the central processing unit of a computer. Equipment which is physically connected to the computer.

Operating system An organized collection of software that controls the overall operations of a computer. The operating system does many basic operations which were performed by hardware in older machines, or which are common to many programs. It is available to the computer at all times either being held in internal storage or on an auxiliary storage device.

Output device A hardware component returns the processed data to user.

P

Parallel Handling all the elements of a word or message simultaneously.

Parity bit An extra bit added to a byte, character, or word, to ensure that there is always either an even number or an odd number of bits, according to the logic of the system.

Pascal A relatively new higher-level programming language that has become increasingly popular on microcomputers because it facilitates the use of good structured programming techniques.

Password A word or code used as a security checkpoint by the system that verifies the users identity.

Peripheral A device—for example, a visual display or printer—used for storing data and entering it into or retrieving it from the computer system.

Plotter An output device which produces line drawings by an automatically controlled pen.

Printer An output device that produces hard copy output.

Procedure The course of action taken for the solution of a problem.

Program A set of coded instructions directing a computer to perform a particular function for the solution of a problem.

Programming language A language used to express computer programs, e.g., BASIC, Pascal or assembly language.

Prompt A character or message provided by the computer to indicate that it's ready to accept keyboard input.

R

Radix The base number in a number system. e.g., the radix in the decimal system is 10.

RAM Random Access Memory. A memory in which each element of information has its own address (location) and from which any element can be easily and conveniently retrieved by using that address. It is the main memory of most microcomputers.

Read To get information from any input or file storage media. For example, reading a magnetic disk by sensing the patterns of magnetism.

Reserved words Certain words which, because they are reserved by operating systems, language

translators, etc. For their own use, cannot be used in an application program.

Resolution The degree of sharpnessof an image, determined by the number of pixels on a screen, expressed as a matrix.

Robot A device equipped with sensing instruments for detecting input signals or environmental conditions, calculating mechanism for making decisions, and guidance mechanism for providing control.

Robotics An area of artificial intelligence related to robots.

ROM Read Only Memory. A solid-state storage chip programmed at the time of its manufacture, and not reprogrammable by the user. Programs stored in ROM are called firmware.

Run The single and continuous execution of a program by a computer on a given set of data.

S

Second generation Computers belonging to the second generation era of technological development of computers when the transistor replaced the vacuum tube. These were prominent from 1959 to 1964, and were displaced by computers using integrated circuitry.

Sector A segmentor division of a track on a disk.

Security Methods of protecting a computer system from improper use or damage.

Serif A typeface with decorative finishing strokes on the tips of the characters; commonly used in the main body of text.

Soft copy Data presented as a video image, in audio format, or in any other form that is not hard copy.

Software A general term for computer programs procedural rules, and the documentation involved in the operation of a computer system.

Source program A computer program written in a source language such as BASIC, FORTRAN, COBOL, Pascal, or assembly language, It is converted into machine language by a special processing program, a compiler, interpreter or assembler.

Statement In programming, an expression or generalized instruction in a source language.

Storage capacity The number of items of data which a storage device is capable of containing. Usually defined in terms of bytes.

Storage device A device used for storing data within a computer system; e.g., integrated circuit storage, magnetic disk unit, magnetic tape unit, magnetic drum unit, floppy disk, tape cassette, etc.

Storage location A position in storage where data and program statements may be stored.

Structured programming An approach or discipline used in the design and coding of computer programs. The approach generally assumes the disciplined use of a few basic coding structures and the use of top-down concepts to decompose main functions into lower-level components for modular coding purposes.

Subroutine A subsidiary routine, within which initial execution never starts. It

is executed when called by some other program, usually the main program.

Subscript A programming notation that is used to identify an element in an array.

Subscripted variable A symbol whose numeric value can change. It is denoted by an array name followed by a subscript; e.g., ART (7).

Supercomputer Computer systems characterized by their very large size and very high processing speeds. Such computers are normally capable of executing many million instructions per second, They are very expensive and are used only for solving very complex problems.

Syntax The grammatical and structural rules of a language. All higher level programming languages possess a formal syntax.

System The computer and all its related components:

Peripheral devices, people, programs, etc.

T

Task A basic unit of work to be accomplished by a computer,

Template A plastic guide used to trace flowcharting symbols.

Temporary storage In programming, storage locations reserved for intermediate results.

Terminal A peripheral device through which information is entered into or extracted from the computer.

Third generation A series of computers that use integrated circuits as their main components.

U

UPC Universal Product Code. A machine-readable code of parallel bars used for labeling grocery products in a supermarket automation system.

User Anyone who utilizes a computer for problem solving or data manipulation.

V

Vacuum tube The dominant electronic element found in computers prior to the advent of the transistor. Computers using vacuum tubes are referred to as first generation computers.

Virus A parasitic program developed as a prank that will damage data, software, or the computer itself.

Voice recognition unit An input device that converts spoken words into machine useable form.

Volatile storage A storage device whose contents are lost if power is removed from the system.

W

Word A group of bits, characters, or bytes considered as an entity and capable of being stored in one storage location.

Word length The number of bits in a computer word.

Word processing A computer system designed to handle words and text as input and output, rather than to perform calculations on numeric values.

Write The process of transferring information from the computer to an output medium.

SOME COMMONLY USED ABBREVIATIONS

Abbreviations	Description
ABC	Atansoff Berry Computer
ADP	Automatic Data processing
AI	Artificial Intelligence
ALGOL	Algorithmic Language
ALU	Arithmetic and Logic Unit
ANSI	American National Standards Institution
API	Application Program Interface
APL	A programming Language
ASCC	Automatic Sequence Controlled Calculator
ASCII	American Standard Code for Information Interchange
ATM	Automated Teller Machine
BASIC	Beginners All-purpose Symbolic Instruction Code
BCD	Binary Coded Decimal
BIOS	Basic Input Output System
CAD	Computer Aided Design
CAI	Computer Aided Instruction
CAL	Computer Aided Learning
CAT	Computer Aided Testing (Also Computer Axial Topography)
CGA	Colour Graphics Adapter
CISC	Complex Instruction Set Computing
CMI	Computer Managed Instruction
COBOL	Common Business Oriented Language
CODEC	COder DECoder
CP	Central Processor
CPU	Central Processing Unit
CPM	Control Program for Microprocessor (Critical Path Method)
CU	Control unit
DAC	Digital to Analog Converter
DBMS	Data Base Management System
DIL or DIP	Dual-In-Line Package
DIMM	Dual-In-Line Memory Module
DOS	Disk Operating System
DRAM	Destructive Read Out
DTP	Desk Top Publishing
EBCIDC	Extended Binary Coded Decimal Interchange Code
EDP	Electronic Data Processing
EEROM	Electrically Erasable ROM
EGA	Enhanced Graphics Adaptor
EOD	End of Data
EOF	End of File
FAT	File Allocation Table
GIGO	Garbage In Garbage Out

HLL	High Level Languages
IAS	Immediate Access Storage
IC	Integrated Circuit
IDP	Integrated Data Processing
IP	Internet Protocol
I/O	Input/Output
IT	Information Technology
K	Kilo (1024)
LAN	Local Area Network
LCD	Liquid Crystal Display
LISP	List Processing
LSI	Large Scale Integration
LSD	Least Significant Bit
MIPS	Million Instructions Per Second
MIS	Management Information System
MSI	Medium Scale Integration
OCR	Optical Character Recognition
OMR	Optical Mark Reading
OS	Operating System
OSI	Open System Interconnection
PC	Personal Computer
PC-AT	Personal Computer with Advanced Technology
PC-XT	Personal Computer with Extended Technology
PCB	Printed Circuit Board
PIPO	Parallel In Parallel Out
PISO	Parallel In Serial Out
PL/I	Programming Language 1
PL/M	Programming Language for Microcomputers
POL	Problem Oriented Language
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RPG	Report Program Generator
SCSI	Small Computer System Interface
SIO	Serial Input/Output
SIP	Single In-line Package
SIMM	Single In-line Memory Module
SQL	Structured Query Language
UNIVAC	Universal Automatic Computer
UPC	Universal Product Code
UVEPROM	Ultra Violet-light Erasable Programmable Read Only Memory
VDU	Visual Display Unit
VLSI	Very Large Scale Integration
WAN	Wide Area Network

Approved by the Federal Ministry of Education (Curriculum Wing, Government of Pakistan, Islamabad vide letter No. F.1-2/2001-Maths (Comp. Sc.) Dated September 23, 2002

National Book Foundation

Islamabad

Rawalpindi-Lahore-Multan-Faisalabad-Wah Cantt-Bahawalpur-Karachi
Hyderabad-Sukkur-Nawabshah-Larkana-Jamshoro-Jacobabad-Peshawar
Abbottabad-Kohat-D.I.Khan-Bannu-Quetta

Price Rs. 109/-